Naval Research Logistics

Single-commodity Stochastic Network Design under Demand and Topological Uncertainties with Insufficient Data

Siqian Shen
Department of Industrial and Operations Engineering
University of Michigan, Ann Arbor, MI 48109, siqian@umich.edu;

Mingdi You
Department of Industrial and Operations Engineering
University of Michigan, Ann Arbor, MI 48109;

Yintai Ma
Department of Industrial Engineering, Tsinghua University, Beijing, China

Abstract: Stochastic network design is fundamental to transportation and logistic problems in practice, yet faces new modeling and computational challenges resulted from heterogeneous sources of uncertainties and their unknown distributions given limited data. In this paper, we design arcs in a network to optimize the cost of single-commodity flows under random demand and arc disruptions. We minimize the network design cost plus cost associated with network performance under uncertainty evaluated by two schemes. The first scheme restricts demand and arc capacities in budgeted uncertainty sets and minimizes the worst-case cost of supply generation and network flows for any possible realizations. The second scheme generates a finite set of samples from statistical information (e.g., moments) of data and minimizes the expected cost of supplies and flows, for which we bound the worst-case cost using budgeted uncertainty sets. We develop cutting-plane algorithms for solving the mixed-integer nonlinear programming reformulations of the problem under the two schemes. We compare the computational efficacy of different approaches and analyze the results by testing diverse instances of random and real-world networks.

Keywords: Two-stage stochastic optimization; robust optimization; mixed-integer linear programming (MILP); linearization techniques; cutting-plane algorithms; valid inequalities

---

1 Introduction

Network design problems (NDPs) arise in many applications that involve service design, construction, and operations. They are of vital importance for building and operating complex systems of telecommunication, energy, transportation, and logistics in the modern society. However, uncertainties associated with these systems may degrade their performance, leading to potential profit losses and service-quality drops. In this paper, we focus on NDPs under both demand and topological uncertainties for operating flows between supply and demand locations. Finite data observations are known for the two uncertainties, but may not be sufficient for deriving exact distributions. We use general statistical information of the given data, including bounds and moments, to construct robust or semi-robust models (explained below in detail) for designing reliable networks.

We consider variants of the single-commodity NDP as follows. A designer builds arcs in a network where each arc has fixed construction cost and capacity. The goal is to minimize a weighted sum of the arc-construction cost and a recourse cost incurred after realizing the two uncertainties. We investigate two problem variants that use different schemes for evaluating the recourse cost, namely, the performance of a designed network under uncertainty. The first one applies robust optimization to handle the issue of unknown distributions, and minimizes the worst-case cost of supply generations and arc flows over two independent uncertainty sets of the demand and arc capacities. This treatment is relatively conservative and assumes that the network designer does not have exact distributional knowledge but only limited statistical information, e.g., bounds on the demand and arc capacity. We refer to the corresponding problem variant as Robust Network Design (RND).

In the second approach, more data related to the uncertain demands and arc disruptions can be obtained by means of simulation, historical experiences, forecasting, etc. From the data we can derive distributions with empirical moments and generate a finite set of samples to formulate a stochastic program, in which we minimize the design cost and the expected cost of supply generation and flows. This model is also embedded with a robust constraint that restricts the worst-case flow cost to be no more than a given threshold over the two budgeted uncertainty sets. We refer to this variant as Semi-Robust Network Design (S-RND).

For both RND and S-RND, we develop cutting-plane algorithms to iteratively optimize their mixed-integer nonlinear programming reformulations. The main objective is to develop effective means for analyzing stochastic NDPs under insufficient data with more than one uncertainty source. Via extensive computational studies based on both randomly generated and real-world networks, we show that S-RND involves little additional computational effort to directly computing a stochastic optimization model, but can guarantee the design robustness with high confidence. S-RND yields lower cost of network design and operations as compared to RND, when more information of the uncertainties are available for generating discrete samples. When only the bounds of the uncertainties are known, RND provides conservative and robust solutions to guarantee high performance under extreme cases when S-RND and the stochastic optimization approach are not applicable.

1.1 Literature Review

NDP studies can be found across a wide range of theoretical and application areas, due to the common NDP structure in many network planning and operation problems in practice. Starting from the study by Magnanti and Wong [39], the literature has tackled NDP with single- and multi-commodity flows [see, e.g., 29]. We refer to the thesis by Poss [48] for a thorough summary of models and algorithms for various NDPs. Meanwhile, other broader classes of NDP include joint location-inventory design [38], transportation-inventory network design [54], facility location design in supply chains [23], road network design [56], and service network design [9].

To consider uncertainties, Lium et al. [37] provide a comprehensive analysis of demand uncertainty in stochastic NDP. Cui et al. [21] optimize facility location design under random arc disruptions, while Mudchanatongsuk et al. [43] analyze both random transportation cost and demand uncertainty. Moreover, metrics other than the expectation have been used to evaluate the network performance under uncertainty, including network reliability (e.g., probability of having unmet demand in supply chains, and probability of having traffic losses in a transportation system) [53, 45, 55], and multiple objectives that balance the cost and risk [18]. Many stochastic NDP studies assume fully known distributions of the uncertainty, and one can formulate the corresponding stochastic programs with finite but large-scale samples. A common approach is the L-shaped method (i.e., the Benders decomposition method) for deriving valid inequalities and iteratively optimizing the large-scale sampling-based reformulations [see, e.g., 47]. Fortz and Poss [24], Botton et al. [16] develop (improved) Benders decomposition approaches for optimizing different NDP models (with multiple layers and constrained hops, respectively). Recently, Crainic et al. [19, 20] propose a scenario decomposition algorithm for stochastic NDP and solve subproblems generated by progressive-hedging heuristics and scenario grouping strategies. Indeed, solving stochastic NDP models requires sufficiently many scenarios to represent the underlying uncertainty. These scenarios can be generated either from true distributions (e.g., the Sample Average Approximation (SAA) method [33]) or statistical information (e.g., moment-matching method [31]).

When the distribution of the uncertainty is unknown and/or the goal is to plan against the worst case, robust optimization is the most popular to model NDP. Altin et al. [2], Koster et al. [34] describe a variety of formulations, complexities, valid inequalities, and computational results for RND variants, mainly with uncertain hose demand. Ukkusuri et al. [59] present robust optimization models that construct arcs or plan arc capacities in transport networks under unknown but bounded demand values. Atamtürk and Zhang [7] propose a two-stage RND with recourse flows under only uncertain demand. They show that the problem is NP-hard even for bipartite network design, and test lot-sizing and location-transportation instances to demonstrate the results as compared to the ones by single-level robust optimization. Under arc capacity uncertainty Minoux [42] show that the related RND is NP-hard in general. Álvarez-Miranda et al. [3] discuss the complexity and heuristic results for single-commodity RND variants; Cacchiani et al. [17] focus on the derivation of optimal solutions to the single-commodity RND by using the branch-and-cut algorithm. They consider uncertainty sets of the hose demand modeled as a finite set of scenarios or as a polytope. Indeed, the RND problem can be modified in a variety of ways, including considering dynamic routing decisions for some specific applications such as in Mattia [40] and Poss and Raack [49]. The resulting optimizing models for RND and its variants are often two-stage mixed-integer linear programs, and can be optimized through decomposition, cutting-plane, and/or column-generation methods [see, e.g., 36, 14, 8, 60].

The theories of stochastic/robust NDP has been substantially applied to optimize performance of a variety of network classes arising in the applications of telecommunication, transportation, and waster distribution [4, 51, 26, 25]. In such contexts, a network designer often faces multi-commodity flows rather than single-commodity flow optimization considered in this paper. Later we demonstrate in Remark 1 that our models can be easily modified to accommodate the multi-commodity setting. However, the extended models require more complex computational approaches. Furthermore, a significant number of NDP studies are closely related to the survivable network analysis [see 28, 32] as well as resilient network design [58]. The literature has studied multicommodity survivable network optimization and focused on advancing solution methods including cutting-plane algorithms [22] and polyhedra studies [57]. In particular, Dahl and Stoer [22] optimize a survivable network design under the uncertainty that any one arc in the network may fail and reformulate the problem using arc installation variables, similar to the feasibility cuts we introduce later in our solution approaches. Atamtürk [5] and Atamtürk and Rajan [6] investigate cut-set and arc-set polyhedron, respectively, to improve the computation of capacitated NDP.

1.2 Contributions

We take into account the distributional ambiguity of two sources of uncertainties, namely, demand and arc-disruption uncertainties involved in network design problems. We first formulate a RND model, of which the robust counterpart is a mixed-integer nonlinear program. We then tailor a semi-robust stochastic optimization model by integrating discrete samples and robust feasibility (i.e., S-RND), which again has a nonlinear reformulation. We optimize both mixed-integer nonlinear programs by using linearization techniques, cutting-plane algorithms, and valid inequalities to achieve fast computation. Through extensive computational studies, we show that the solution given by S-RND is not as conservative and costly as the one given by RND if more information about the uncertainties are given other than just the lower and upper bounds so that we can derive discrete samples used in S-RND. Meanwhile, comparing with a pure stochastic optimization model based on the same set of limited uncertainty samples, the S-RND model provides a more reliable design by only slightly increasing the computational time.

1.3 Structure of the Paper

The remainder of the paper is organized as follows. Section 2 describes a generic model for both RND and S-RND. Section 3 and Section 4 specify the two models and develop cutting-plane algorithms for each problem, respectively. Section 5 derives additional valid inequalities based on special structures of node degrees and arc capacities. Section 6 tests the approaches for solving RND and S-RND on diverse network instances, describes computational results, and provides solution analysis. Section 7 concludes the paper and states future research.

2 Problem Description and Formulation Overview

Let $G(\mathcal{N},\mathcal{A})$ be a directed connected graph with the node set $\mathcal{N}=\mathcal{N}_+\cup\mathcal{N}_=\cup\mathcal{N}_-$, where $\mathcal{N}_+,\mathcal{N}_=$ and $\mathcal{N}_-$ respectively represent sets of supply, transmission, and demand nodes, satisfying $\mathcal{N}_+\cap\mathcal{N}_= = \mathcal{N}_+\cap\mathcal{N}_- = \mathcal{N}_-\cap\mathcal{N}_= = \emptyset$. The set $\mathcal{A}\subset\mathcal{N}\times\mathcal{N}$ includes all arcs that can be potentially constructed. Associated with each arc $(i,j)\in\mathcal{A}$ are the construction cost $c_{ij}>0$, flow capacity $a_{ij}>0$, and unit flow cost $d_{ij}>0$. Let h_i be the unit cost of supply generation, $S_i\geq 0$ be the generation capacity at node i, $\forall i\in\mathcal{N}_+$, and $D_i\geq 0$ be the random demand at node i, $\forall\in\mathcal{N}_-$. Define binary variables x_{ij}, for all $(i,j)\in\mathcal{A}$, such that $x_{ij}=1$ if we construct arc (i,j), and 0 otherwise. For each $(i,j)\in\mathcal{A}$, define variable $f_{ij}\geq 0$ as the amount of flow on arc (i,j), and for each supply node $i\in\mathcal{N}_+$, define variable $g_i\geq 0$ as the amount of supply generated at node $i\in\mathcal{N}_-$. To model arc disruption, we introduce a random vector $\mathbf{I}=[I_{ij},\ (i,j)\in\mathcal{A}]^\mathsf{T}\in[0,1]^{|\mathcal{A}|}$, bounded within an uncertainty set U_I, with each I_{ij} representing the remaining percentage of the capacity at arc (i,j), $\forall(i,j)\in A$ after some random disruptions. The random demand vector $\mathbf{D}=[D_i,\ i\in\mathcal{N}_-]^\mathsf{T}$ is bounded in an uncertainty set U_D. We specify the two uncertainty sets U_I and U_D as "budgeted uncertainty sets" [see, e.g., 12] described in Section 3.1.

Denote $\mathbf{x}$ as the vector form of variables x_{ij}, $\forall(i,j)\in\mathcal{A}$. Let $V(\mathbf{x},\mathbf{D},\mathbf{I})$ be the minimum flow cost given arc construction decision $\mathbf{x}$, demand $\mathbf{D}$, and disruption $\mathbf{I}$.

$$
\begin{aligned}
V(\mathbf{x},\mathbf{D},\mathbf{I}) = \quad \min_{\mathbf{f},\mathbf{g}} \quad & \sum_{i\in\mathcal{N}_+} h_i g_i + \sum_{(i,j)\in\mathcal{A}} d_{ij} f_{ij} && \text{(1a)}\\
\text{s.t.} \quad & \sum_{j:(i,j)\in\mathcal{A}} f_{ij} - \sum_{j:(j,i)\in\mathcal{A}} f_{ji} - g_i = 0 \quad \forall i\in\mathcal{N}_+ && \text{(1b)}\\
& \sum_{j:(i,j)\in\mathcal{A}} f_{ij} - \sum_{j:(j,i)\in\mathcal{A}} f_{ji} = -D_i \quad \forall i\in\mathcal{N}_- && \text{(1c)}\\
& \sum_{j:(i,j)\in\mathcal{A}} f_{ij} - \sum_{j:(j,i)\in\mathcal{A}} f_{ji} = 0 \quad \forall i\in\mathcal{N}_= && \text{(1d)}\\
& 0\leq g_i\leq S_i \quad \forall i\in\mathcal{N}_+ && \text{(1e)}\\
& 0\leq f_{ij}\leq a_{ij} I_{ij} x_{ij} \quad \forall (i,j)\in\mathcal{A}. && \text{(1f)}
\end{aligned}
$$

In the objective (1a), $\sum_{i\in\mathcal{N}_+} h_i g_i+\sum_{(i,j)\in\mathcal{A}} d_{ij} f_{ij}$ is the sum of flow cost and supply generation cost on the remaining network after disruption. Constraints (1b)–(1d) model the flow balance at nodes in $\mathcal{N}_+$, $\mathcal{N}_-$ and $\mathcal{N}_=$, respectively. Constraints (1e) bound variables g_i from above by S_i, $\forall i\in\mathcal{N}_+$. Constraints (1f) allow flow f_{ij} being positive when $x_{ij}=1$ and $I_{ij}>0$, meaning that arc (i,j) has been constructed and the remaining arc capacity after disruption is positive.

We consider a two-stage structure for both RND and S-RND problems as follows. At the first stage, we build arc capacities, i.e., decide the values of $x_{ij},\ \forall(i,j)\in\mathcal{A}$, before realizing uncertain demand and arc disruptions. Both decision vectors $\mathbf{f}$ and $\mathbf{g}$ are recourse decisions at the second stage, and we formulate the second-stage objective by measuring $V(\mathbf{x},\mathbf{D},\mathbf{I})$ for given first-stage decision vector $\mathbf{x}$, and parameters $\mathbf{D}$ and $\mathbf{I}$. The goal is to minimize a weighted sum of cost objectives from both stages. A generic form of the two problem variants reads:

$$\min_{\mathbf{x}\in\mathbf{X}} \quad \rho\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho)y, \tag{2}$$

where $\mathbf{X}\subseteq\{0,1\}^{|\mathcal{A}|}$ represents a feasible region, consisting of constraints to ensure that $\mathbf{x}$ satisfies supply and demand at all nodes, given any possible arc disruption and uncertain demand. Parameter $\rho\in[0,1]$ is a weight related to the designer's tradeoff preference between the first-stage construction cost $\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij}$ and the second-stage recourse cost $y\in\mathbb{R}_+$.

We calculate y based on two criteria for evaluating the random cost $V(\mathbf{x},\mathbf{D},\mathbf{I})$. First, we follow a robust optimization scheme and minimize the cost of network flows and supply generations in the worst case (i.e., $y=\max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I})$). For a given $\mathbf{x}$, $V(\mathbf{x},\mathbf{D},\mathbf{I})$ in (1) may be infeasible for some realized $\mathbf{I}$ or $\mathbf{D}$. Denote $\mathcal{R}_x$ as the feasible region of arcs constructed at the first stage to be robust with respect to all possible arc disruption and demand, i.e. given $\mathbf{x}\in\mathcal{R}_x$, Problem (1) is always feasible for any $\mathbf{I}\in U_I$ and $\mathbf{D}\in U_D$. We design a robust network by solving

$$\textbf{RND:} \quad \min_{\mathbf{x}\in\mathcal{R}_x} \quad \rho\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho)\max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I}). \tag{3}$$

Alternatively, a stochastic programming scheme [cf. 15] considers $y=E_\xi[V(\mathbf{x},\mathbf{D}_\xi,\mathbf{I}_\xi)]$ as the expected cost of supply generations and flows, given random parameters $\mathbf{D}_\xi$ and $\mathbf{I}_\xi$, i.e., we can optimize $=\min_{\boldsymbol{x}}\rho\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij}+(1-\rho)E_\xi[V(\mathbf{x},\mathbf{D}_\xi,\mathbf{I}_\xi)]$. However, when sampling a large number of scenarios, the computation is generally inefficient. Also, a fully known distribution of ξ may not be available due to various issues such as the difficulty of data collection in highly uncertain environments. Here we explore an alternative approach as follows. We generate a finite set of scenarios, from statistical information of the uncertainties $\mathbf{D}_\xi$ and $\mathbf{I}_\xi$ that one can be derived from given data. We optimize a two-stage stochastic program formulated by using the scenarios, in which we also require certain level of solution robustness with respect to uncertainty sets of demands and arc disruptions deduced from existing data.

Let $(\mathbf{I}^1,\mathbf{D}^1),(\mathbf{I}^2,\mathbf{D}^2),\ldots,(\mathbf{I}^N,\mathbf{D}^N)$ be an independently and identically distributed (i.i.d) samples of the random vectors $\mathbf{I},\mathbf{D}$. The corresponding sample average function is $\frac{1}{N}\sum_{\omega=1}^{N} V(\mathbf{x},\mathbf{D}^\omega,\mathbf{I}^\omega)$. As the sample average function cannot reflect all possible values of $\mathbf{I}$ and $\mathbf{D}$, we impose the following two types of solution robustness: First, solution $\mathbf{x}$ should be feasible for all realizations from the uncertainty sets U_I and U_D. Second, the maximum possible $V(\mathbf{x},\mathbf{D},\mathbf{I})$ for all $\mathbf{I}\in U_I,\mathbf{D}\in U_D$ is bounded by an upper bound $\mathcal{L}$. We design a semi-robust network by solving

$$\min_{\mathbf{x}\in\mathcal{S}_x} \quad \rho\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho)E_\xi[V(\mathbf{x},\mathbf{D}_\xi,\mathbf{I}_\xi)], \tag{4}$$

where feasible region $\mathcal{S}_x$ is described as

$$\mathcal{S}_x = \left\{\mathbf{x}\in\mathcal{R}_x \;\middle|\; \max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I})\le\mathcal{L}\right\}. \tag{5}$$

We consider S-RND as the following scenario-based formulation:

$$\textbf{S-RND:} \quad \min_{\mathbf{x}\in\mathcal{S}_x} \ \rho\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho)\sum_{\omega=1}^{N} V(\mathbf{x},\mathbf{D}^\omega,\mathbf{I}^\omega)/N, \tag{6}$$

Let $\mathcal{V}^*$ denote the optimal objective value of Problem (4), and $\mathcal{X}^\epsilon$ denote ϵ-optimal solution set of Problem (4), i.e., if $\bar{\mathbf{x}} \in \mathcal{X}^\epsilon$ then $\rho \sum_{(i,j)\in\mathcal{A}} c_{ij}\bar{x}_{ij} + (1-\rho)E_\xi[V(\bar{\mathbf{x}}, \mathbf{D}_\xi, \mathbf{I}_\xi)] \le \mathcal{V}^* + \epsilon$. Similarly, we define $\hat{\mathcal{V}}_N^*$ and $\hat{\mathcal{X}}_N^\epsilon$ to be the optimal objective value of Problem (6) and ϵ-optimal solution set of Problem (6), respectively.

Proposition 1. [33] (i) As $N \to \infty$, $\hat{\mathcal{V}}_N^* \to \mathcal{V}^*$ with probability 1; (ii) for any $\epsilon \ge 0$, the event $\{\hat{\mathcal{X}}_N^\epsilon \subset \mathcal{X}^\epsilon\}$ happens with probability 1 as $N \to \infty$, and the probability of the event $\hat{\mathcal{X}}_N^\epsilon \subset \mathcal{X}^\epsilon$ approaches 1 exponentially fast as $N \to \infty$.

Proof. The feasible region of Problem (4) and Problem (6) are finite because $\mathbf{x} \in \{0,1\}^{|\mathcal{A}|}$. According to the definition of $\mathcal{R}_x$, for any $\mathbf{D} \in U_D, \mathbf{I} \in U_I$ and $\mathbf{x} \in \mathcal{R}_x$, $V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ has finite values. Thus, $E_\xi[V(\mathbf{x}, \mathbf{D}_\xi, \mathbf{I}_\xi)] < \infty$ for any $\mathbf{x} \in \mathcal{R}_x$. Therefore, the two properties directly follow from Proposition 2.1 and Proposition 2.2 in Kleywegt et al. [33]. □

Proposition 1 guarantees asymptotical convergence by using the scenario-based formulation to approximate the true solution of S-RND.

3 Models and Algorithms of RND

For RND, we solve a relaxed master problem at the first stage:

$$\begin{aligned}
\mathbf{MP1}: \quad \min \quad & \rho \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho)y && \text{(7a)}\\
\text{s.t.} \quad & L_1(\mathbf{x}) \ge 0 && \text{(7b)}\\
& L_2(y, \mathbf{x}) \ge 0 && \text{(7c)}\\
& x_{ij} \in \{0,1\} \ \forall (i,j) \in \mathcal{A}, \quad y \ge 0, && \text{(7d)}
\end{aligned}$$

where y equals to $\max_{\mathbf{I}\in U_I, \mathbf{D}\in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I})$, $\forall \mathbf{x} \in \mathcal{R}_x$. Set $L_1(\mathbf{x}) \ge 0$ consists of valid feasibility cuts, and set $L_2(y, \mathbf{x}) \ge 0$ consists of valid optimality cuts. We describe the methodological details of deriving these two types of cuts in Section 3.2 and Section 3.3, respectively.

3.1 Budgeted Uncertainty Sets

For demand uncertainty, let D_i, $\forall i \in \mathcal{N}_-$ be a random variable with mean $\bar{D}_i$, lower bound $\bar{D}_i - \tilde{D}_i$, and upper bound $\bar{D}_i + \tilde{D}_i$, i.e., $D_i \in [\bar{D}_i - \tilde{D}_i, \bar{D}_i + \tilde{D}_i]$, $\forall i \in \mathcal{N}_-$. (Assume that $\bar{D}_i - \tilde{D}_i \ge 0$, $\forall i \in \mathcal{N}_-$.) We formulate a budget constraint $\sum_{i\in\mathcal{N}_-} \pi_i D_i \le \Pi$, where Π and π_i, $\forall i \in \mathcal{N}_-$ are fixed parameter. The budgeted demand uncertainty set is

$$U_D = \left\{ \mathbf{D} \in \mathbb{R}_+^{|\mathcal{N}_-|} \;\middle|\; \sum_{i\in\mathcal{N}_-} \pi_i D_i \le \Pi,\ D_i \in [\bar{D}_i - \tilde{D}_i, \bar{D}_i + \tilde{D}_i],\ \forall i \in \mathcal{N}_- \right\}. \tag{8}$$

For arc disruption uncertainty, let Γ be a given parameter that can be viewed as the maximum sum of percentage of failed arcs allowed in any realizations. The budgeted uncertainty set of arc disruption is

$$U_I = \left\{ \mathbf{I} \in [0,1]^{|\mathcal{A}|} \;\middle|\; \sum_{(i,j)\in\mathcal{A}} I_{ij} \ge |\mathcal{A}| - \Gamma \right\}. \tag{9}$$

3.2 A Separation Problem and Valid Inequalities for Defining $\mathcal{R}_x$

We specify feasibility cuts in $L_1(\mathbf{x}) \geq 0$ that provide an exact description of $\mathcal{R}_x$.

Theorem 1. Given binary solutions $x_{ij},\ \forall (i,j) \in \mathcal{A}$, decision vector $\mathbf{x} \in \mathcal{R}_x$ if and only if

$$\sum_{i \in \mathcal{N}_+ \cap \widetilde{\mathcal{N}}} S_i - \sum_{i \in \mathcal{N}_- \cap \widetilde{\mathcal{N}}} D_i + \sum_{(i,j) \in \mathcal{A} \cap \phi^+(\widetilde{\mathcal{N}})} a_{ij} x_{ij} I_{ij} \geq 0, \quad \forall\, \mathbf{I} \in U_I,\ \mathbf{D} \in U_D,\ \widetilde{\mathcal{N}} \subseteq \mathcal{N}. \tag{10}$$

where for any $\widetilde{\mathcal{N}} \subseteq \mathcal{N}$, set $\phi^+(\widetilde{\mathcal{N}})$ includes all the incoming arcs of the nodes in $\widetilde{\mathcal{N}}$.

This result directly follows the Gale-Hoffman inequalities [27, 30], for which we provide a proof using Farkas Lemma in Appendix A. The proof does not require special types of uncertainty sets U_D and U_I, and the result can be generalized to any uncertainty sets rather than the budgeted uncertainty considered in this paper. One can change parameters Π of U_D and Γ of U_I to enlarge or strengthen the feasible region $\mathbf{x} \in \mathcal{R}_x$ defined by inequalities (10). Specifically, larger Π- and Γ-values will enlarge sets U_D and U_I and thus require the solution $\mathbf{x}$ satisfying (10) to be more robust. We can also use different U_D, U_I in (10) and in $\max_{\mathbf{I} \in U_I, \mathbf{D} \in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ by changing their corresponding Π and Γ. Constraint (10) seeks feasible $\mathbf{x}$ with respect to any possible realizations of uncertain $\mathbf{D}$ and $\mathbf{I}$, and $\max_{\mathbf{I} \in U_I, \mathbf{D} \in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ further examines the worst-case objective value under certain uncertainty sets that can be different from the ones used to enforce $\mathbf{x} \in \mathcal{R}_x$. One can also vary the upper and lower bounds of D_i for some $i \in \mathcal{N}_-$ in set U_D to reflect different levels of feasibility strictness.

According to Theorem 1, given $x_{ij} \in \{0,1\},\ \forall (i,j) \in \mathcal{A}$, we formulate a separation problem to check whether x_{ij} belongs to $\mathcal{R}_x$, i.e., whether $\mathbf{x}$ satisfies (10) for any $\mathbf{D} \in U_D, \mathbf{I} \in U_I$, and $\widetilde{\mathcal{N}} \subseteq \mathcal{N}$. The separation problem is given by

$$\begin{aligned}
\mathrm{SP}_{\mathcal{R}}:\ R(\mathbf{x}) = \quad \min_{\mathbf{v},\mathbf{w},\mathbf{I},\mathbf{D}} \quad & \sum_{i \in \mathcal{N}_+} S_i v_i - \sum_{i \in \mathcal{N}_-} D_i v_i + \sum_{(i,j) \in \mathcal{A}} a_{ij} x_{ij} I_{ij} w_{ij} && \text{(11a)}\\
\text{s.t.} \quad & v_j - v_i \leq w_{ij} \quad \forall (i,j) \in \mathcal{A} && \text{(11b)}\\
& v_i \in \{0,1\}\ \forall i \in \mathcal{N}, \quad w_{ij} \in \{0,1\}\ \forall (i,j) \in \mathcal{A} && \text{(11c)}\\
& \sum_{i \in \mathcal{N}_-} \pi_i D_i \leq \Pi && \text{(11d)}\\
& \bar{D}_i - \tilde{D}_i \leq D_i \leq \bar{D}_i + \tilde{D}_i \quad \forall i \in \mathcal{N}_- && \text{(11e)}\\
& \sum_{(i,j) \in \mathcal{A}} I_{ij} \geq |\mathcal{A}| - \Gamma && \text{(11f)}\\
& 0 \leq I_{ij} \leq 1 \quad \forall (i,j) \in \mathcal{A} && \text{(11g)}
\end{aligned}$$

where $R(\mathbf{x})$ is the minimum objective value of $\mathrm{SP}_{\mathcal{R}}$ given any solution $\mathbf{x}$. In $\mathrm{SP}_{\mathcal{R}}$, $v_i = 1$ if and only if $i \in \widetilde{\mathcal{N}} \subseteq \mathcal{N}$. Due to (11b), w_{ij} is forced to be 1 if arc $(i,j) \in \phi^+(\widetilde{\mathcal{N}})$. This is because $a_{ij} x_{ij} I_{ij} \geq 0$ and we minimize (11a), then optimal solutions w_{ij} to $\mathrm{SP}_{\mathcal{R}}$ will identify a cut of $\phi^+(\widetilde{\mathcal{N}})$ and solutions v_i correspond to a subset $\widetilde{\mathcal{N}} \subseteq \mathcal{N}$. Constraints (11d), (11e), (11f), (11g) define sets U_I and U_D.

Given $\mathbf{x}$, we optimize $\mathrm{SP}_{\mathcal{R}}$ by solving Problem (11). Letting $v_i = 0,\ \forall i \in \mathcal{N}$, $w_{ij} = 0,\ \forall (i,j) \in \mathcal{A}$, we can trivially satisfy all constraints and obtain a feasible objective value 0. Therefore, $R(\mathbf{x})$ must be 0 or less. If $R(\mathbf{x}) = 0$, (10) holds for any $\mathbf{D} \in U_D$, $\mathbf{I} \in U_I$, and $\widetilde{\mathcal{N}} \subseteq \mathcal{N}$, which verifies $\mathbf{x} \in \mathcal{R}_x$. Otherwise if $R(\mathbf{x}) < 0$, then there exist $\mathbf{D}' \in U_D$ and $\mathbf{I}' \in U_I$, such that $V(\mathbf{x}, \mathbf{D}', \mathbf{I}')$ has no feasible solutions $g_i \geq 0,\ \forall i \in \mathcal{N}_+$ and $f_{ij} \geq 0,\ \forall (i,j) \in \mathcal{A}$. We then generate a feasibility cut into $L_1(\mathbf{x}) \geq 0$ to **MP1** to exclude such a solution $\mathbf{x}$.

We further linearize $D_i v_i$ and $I_{ij} w_{ij}$ in $\mathrm{SP}_{\mathcal{R}}$ using McCormick Inequalities [41]. Letting $D_i v_i \equiv b_i,\ i \in \mathcal{N}_-$ and $I_{ij} w_{ij} \equiv p_{ij},\ (i,j) \in \mathcal{A}$, we reformulate $\mathrm{SP}_{\mathcal{R}}$ as an equivalent mixed-integer linear programming (MILP) model:

$$\begin{aligned}
\mathrm{SP}^{\mathrm{MILP}}_{\mathcal{R}}: \quad \min \quad & \sum_{i\in\mathcal{N}_+} S_i v_i - \sum_{i\in\mathcal{N}_-} b_i + \sum_{(i,j)\in\mathcal{A}} a_{ij} x_{ij} p_{ij} && \text{(12a)}\\
\text{s.t.} \quad & \text{(11b), (11c), (11d), (11e), (11f), (11g)} &&\\
& I_{ij} + w_{ij} - 1 - p_{ij} \le 0 \quad \forall (i,j) \in \mathcal{A} && \text{(12b)}\\
& p_{ij} \ge 0 \quad \forall (i,j) \in \mathcal{A} && \text{(12c)}\\
& b_i - (\bar{D}_i + \tilde{D}_i) v_i \le 0 \quad \forall i \in \mathcal{N}_- && \text{(12d)}\\
& b_i - D_i \le 0 \quad \forall i \in \mathcal{N}_-, && \text{(12e)}
\end{aligned}$$

where McCormick inequalities $p_{ij} \le w_{ij}$, $p_{ij} \le I_{ij}$, and $p_{ij} \le 1$, for all $(i,j) \in \mathcal{A}$ are satisfied automatically due to the minimization objective, and thus are omitted in (12). To see that $\mathrm{SP}^{\mathrm{MILP}}_{\mathcal{R}}$ is equivalent to $\mathrm{SP}_{\mathcal{R}}$ in (11): if $w_{ij} = 1$, since we minimize $\sum_{(i,j)\in\mathcal{A}} a_{ij} x_{ij} p_{ij}$ and $a_{ij} x_{ij} \ge 0$, then $p_{ij} = I_{ij}$ due to (12b); if $v_i = 0$, due to the minimization of $-\sum_{i\in\mathcal{N}_-} b_i$, we have $b_i = 0$ according to (12d); if $v_i = 1$, then $b_i = D_i$ due to (12e).

If $\mathbf{x} \notin \mathcal{R}_x$, then there exists an optimal solution $(\hat{v}_i, \hat{b}_i, \hat{p}_{ij})$ to Problem $\mathrm{SP}^{\mathrm{MILP}}_{\mathcal{R}}$ in (12) such that $\sum_{i\in\mathcal{N}_+} S_i \hat{v}_i - \sum_{i\in\mathcal{N}_-} \hat{b}_i + \sum_{(i,j)\in\mathcal{A}} a_{ij} x_{ij} \hat{p}_{ij} < 0$. Correspondingly, we generate a feasibility cut to $L_1(\mathbf{x}) \ge 0$ as:

$$\sum_{i\in\mathcal{N}_+} S_i \hat{v}_i - \sum_{i\in\mathcal{N}_-} \hat{b}_i + \sum_{(i,j)\in\mathcal{A}} a_{ij} \hat{p}_{ij} x_{ij} \ge 0. \tag{13}$$

Theorem 2. Cut (13) is valid for all $\mathbf{x} \in \mathcal{R}_x$.

Proof. Note that $(\hat{v}_i, \hat{b}_i, \hat{p}_{ij})$ is a feasible solution to (12). If $\mathbf{x} \in \mathcal{R}_x$, then the corresponding objective value $\sum_{i\in\mathcal{N}_+} S_i \hat{v}_i - \sum_{i\in\mathcal{N}_-} \hat{b}_i + \sum_{(i,j)\in\mathcal{A}} a_{ij} \hat{p}_{ij} x_{ij}$ must be greater or equal to 0. This shows the validity of Cut (13). □

3.3 Optimality Cuts

We now derive optimality cuts $L_2(y, \mathbf{x}) \ge 0$ by considering the dual of $V(\mathbf{x}, \mathbf{D}, \mathbf{I})$, formulated as

$$\begin{aligned}
\max \quad & \sum_{i\in\mathcal{N}_-} -D_i \mu_i + \sum_{i\in\mathcal{N}_+} S_i \nu_i + \sum_{(i,j)\in\mathcal{A}} a_{ij} x_{ij} I_{ij} \gamma_{ij} && \text{(14a)}\\
\text{s.t.} \quad & -\mu_i + \nu_i \le h_i \quad \forall i \in \mathcal{N}_+ && \text{(14b)}\\
& \mu_i - \mu_j + \gamma_{ij} \le d_{ij} \quad \forall (i,j) \in \mathcal{A} && \text{(14c)}\\
& \nu_i \le 0 \quad \forall i \in \mathcal{N}_+ && \text{(14d)}\\
& \gamma_{ij} \le 0 \quad \forall (i,j) \in \mathcal{A} && \text{(14e)}
\end{aligned}$$

where dual variables μ_i, $\nu_i \le 0$, $\gamma_{ij} \le 0$ are associated with constraints (1b)–(1d), (1e), and (1f), respectively. Given that the dual is a maximization problem, we have

$$\max_{\mathbf{I}\in U_I, \mathbf{D}\in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I}) = \max\{\text{(14a)} : \text{(14b)–(14e), (11d)–(11g)}\} \tag{15}$$

which is a bilinear program. Unlike $\mathrm{SP}_{\mathcal{R}}$, all the dual variables in model (15) are not necessarily binary, and thus we cannot directly replace the bilinear terms $D_i \mu_i$ and $I_{ij} \gamma_{ij}$ with additional variables and linear constraints. Next, we explore characteristics of optimal solutions to model (15), and reformulate (15) as an MILP model.

Proposition 2. When Γ is integer, all extreme points of U_I are binary valued.

Proof. For any feasible $\mathbf{I} \in U_I$ given integer parameter Γ, if it contains a fractional component, then there are not enough linearly independent constraints tight at this solution and thus it cannot correspond to an extreme point. $\square$

Proposition 3. When Γ is integer, there exists an optimal solution to (15) that has all I_{ij}, $\forall(i,j) \in \mathcal{A}$ binary valued.

Proof. For a bilinear program, a global optimum can be found among pairs of extreme points from respective feasible regions [e.g., 44]. Thus, at least one extreme point of U_I, which has all I_{ij}, $\forall(i,j) \in \mathcal{A}$ binary valued according to Proposition 2, optimizes (15). $\square$

Given the above result, we can generalize all the approaches in this paper to handle the variant with complete 0-1 arc closure, i.e., $\mathbf{I} \in \{0,1\}^{|\mathcal{A}|}$. For integer Γ, the linearization steps remain the same due to the fact that the optimal solutions will always resolve at binary realizations of $\mathbf{I}$. Next, we linearize $D_i\mu_i$ according to the following properties of any optimal solution.

Proposition 4. When $\sum_{i\in\mathcal{N}_-} \pi_i(\bar{D}_i + \tilde{D}_i) \geq \Pi$, there exists an optimal $\mathbf{D} \in U_D$ to the mixed-integer bilinear programming model (15) satisfying the following two properties: (i) all $D_i, i \in \mathcal{N}_-$ either equals to $\bar{D}_i - \tilde{D}_i$ or $\bar{D}_i + \tilde{D}_i$ except for at most one D_i; (ii) $\sum_{i\in\mathcal{N}_-} \pi_i D_i = \Pi$.

Proof. Similar to Proposition 3, at least one extreme point of U_D optimizes the bilinear program. Considering extreme points, or equivalently, the basic feasible solutions of U_D, it is easy to see that at least $|\mathcal{N}_-|-1$ inequalities of $\bar{D}_i - \tilde{D}_i \leq D_i \leq \bar{D}_i + \tilde{D}_i$, $\forall i \in \mathcal{N}_-$ have zero slacks, showing that at least $|\mathcal{N}_-|-1$ of $D_i, i \in \mathcal{N}_-$ either equal to $\bar{D}_i - \tilde{D}_i$ or $\bar{D}_i + \tilde{D}_i$. To see $\sum_{i\in\mathcal{N}_-} \pi_i D_i = \Pi$, we verify the result by contradiction: Suppose that $D_i^\star, \forall i \in \mathcal{N}_-$ are optimal to (15) with $\sum_{i\in\mathcal{N}_-} \pi_i D_i^\star < \Pi$, then for any $D_{i'}^\star < \bar{D}_{i'}^\star + \tilde{D}_{i'}^\star$, we can increase its value until either it reaches the upper bound, or $\sum_{i\in\mathcal{N}_-} \pi_i D_i = \Pi$. Such a modification keeps the solution feasible without decreasing the objective value until $\sum_{i\in\mathcal{N}_-} \pi_i D_i = \Pi$. This completes our proof. $\square$

We now propose an MILP reformulation of (15) under the assumption that parameters Γ and Π in the budgeted uncertainty sets are both integral. According to Proposition 3 and Proposition 4, model (15) can be solved by computing $|\mathcal{N}_-|$ subproblems, each of which sets one D_i, $i \in \mathcal{N}_-$ in between its upper bound $(\bar{D}_i+\tilde{D}_i)$ and lower bound $(\bar{D}_i-\tilde{D}_i)$. Therefore, in the k^{th} subproblem, we designate $D_k = \left(\Pi - \sum_{i\in\mathcal{N}_-\setminus\{k\}} \pi_i D_i\right)/\pi_k$ with other D_i variables either being $\bar{D}_i - \tilde{D}_i$ or $\bar{D}_i + \tilde{D}_i$, $\forall i \neq k$. We add superscript (k) to all variables in model (15), and present the k^{th} subproblem as

$$\max \quad \sum_{i\in\mathcal{N}_-\setminus\{k\}} D_i^{(k)}\left(-\mu_i^{(k)} + \frac{\pi_i}{\pi_k}\mu_k^{(k)}\right) - \frac{\Pi}{\pi_k}\mu_k^{(k)} + \sum_{i\in\mathcal{N}_+} S_i\nu_i^{(k)} + \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij}I_{ij}^{(k)}\gamma_{ij}^{(k)} \tag{16a}$$

$$\text{s.t.} \quad \text{(14b)–(14e),(11f)}$$

$$D_i^{(k)} \in \{\bar{D}_i - \tilde{D}_i, \bar{D}_i + \tilde{D}_i\} \quad \forall i \in \mathcal{N}_-\setminus\{k\} \tag{16b}$$

$$I_{ij}^{(k)} \in \{0,1\} \quad \forall(i,j) \in \mathcal{A}, \tag{16c}$$

where (16b) indicate that $D_i^{(k)}$ either equals to $\bar{D}_i - \tilde{D}_i$ or $\bar{D}_i + \tilde{D}_i$, $\forall i \in \mathcal{N}_-\setminus\{k\}$. We then define binary variables $z_i^{(k)} \in \{0,1\}$, such that $z_i^{(k)} = 0$ if $D_i^{(k)} = \bar{D}_i - \tilde{D}_i$, and $z_i^{(k)} = 1$ if $D_i^{(k)} = \bar{D}_i + \tilde{D}_i$,

$\forall i \in \mathcal{N}_-\backslash\{k\}$. With both variables $z_i^{(k)}$ and $I_{ij}^{(k)}$ being binary, we can linearize subproblem (16) for each $k \in \mathcal{N}_-$ and obtain

$$\max \quad \sum_{i\in\mathcal{N}_-\backslash\{k\}} \alpha_i^{(k)} - \frac{\Pi}{\pi_k}\mu_k^{(k)} + \sum_{i\in\mathcal{N}_+} S_i\nu_i^{(k)} + \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij}\sigma_{ij}^{(k)} \tag{17a}$$

$$\text{s.t.} \quad \text{(14b)–(14e),(11f),(16c)}$$

$$\alpha_i^{(k)} \leq \left(\bar{D}_i + \tilde{D}_i\right)\left(-\mu_i^{(k)} + \frac{\pi_i}{\pi_k}\mu_k^{(k)}\right) + M^1\left(1 - z_i^{(k)}\right) \quad \forall i \in \mathcal{N}_-\backslash\{k\} \tag{17b}$$

$$\alpha_i^{(k)} \leq \left(\bar{D}_i - \tilde{D}_i\right)\left(-\mu_i^{(k)} + \frac{\pi_i}{\pi_k}\mu_k^{(k)}\right) + M^1 z_i^{(k)} \quad \forall i \in \mathcal{N}_-\backslash\{k\} \tag{17c}$$

$$z_i^{(k)} \in \{0,1\} \quad \forall i \in \mathcal{N}_-\backslash\{k\} \tag{17d}$$

$$\sigma_{ij}^{(k)} \leq \gamma_{ij}^{(k)} + M^2\left(1 - I_{ij}^{(k)}\right) \quad \forall (i,j) \in \mathcal{A} \tag{17e}$$

$$\sigma_{ij}^{(k)} \leq 0 \quad \forall (i,j) \in \mathcal{A} \tag{17f}$$

To see the equivalence of (17) and (16), in (17b) and (17c), M^1 is arbitrarily large and the objective maximizes $\sum_{i\in\mathcal{N}_-\backslash\{k\}} \alpha_i^{(k)}$. Thus, either $\alpha_i^{(k)}$ equals to $\left(\bar{D}_i + \tilde{D}_i\right)\left(-\mu_i^{(k)} + \pi_i\mu_k^{(k)}/\pi_k\right)$ if $z_i^{(k)} = 1$, indicating that $D_i^{(k)} = \bar{D}_i + \tilde{D}_i$, or $\alpha_i^{(k)}$ equals to $\left(\bar{D}_i - \tilde{D}_i\right)\left(-\mu_i^{(k)} + \pi_i\mu_k^{(k)}/\pi_k\right)$ if $z_i^{(k)} = 0$, indicating that $D_i^{(k)} = \bar{D}_i - \tilde{D}_i$. Similarly, in (17e) and (17f), because $a_{ij}x_{ij} \geq 0$, $\gamma_{ij}^{(k)} \leq 0$ and we maximize $\sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij}\sigma_{ij}^{(k)}$, if $I_{ij}^{(k)} = 1$, then $\sigma_{ij}^{(k)}$ equals to $\gamma_{ij}^{(k)}$; otherwise if $I_{ij}^{(k)} = 0$, constraints (17e) are relaxed given an arbitrary large M^2, and $\sigma_{ij}^{(k)}$ equals to 0 due to (17f).

Derivation of M^1 and M^2: To tighten the MILP formulation (17), we derive lower bounds of M^1 and M^2, which need to guarantee the validity of their related constraints in (17). From (17b) and (17c), M^1 must satisfy

$$\left(\bar{D}_i + \tilde{D}_i\right)\left(-\mu_i^{(k)} + \frac{\pi_i}{\pi_k}\mu_k^{(k)}\right) + M^1 \geq \left(\bar{D}_i - \tilde{D}_i\right)\left(-\mu_i^{(k)} + \frac{\pi_i}{\pi_k}\mu_k^{(k)}\right), \quad \forall i \in \mathcal{N}_-\backslash\{k\} \tag{18a}$$

$$\left(\bar{D}_i + \tilde{D}_i\right)\left(-\mu_i^{(k)} + \frac{\pi_i}{\pi_k}\mu_k^{(k)}\right) \leq \left(\bar{D}_i - \tilde{D}_i\right)\left(-\mu_i^{(k)} + \frac{\pi_i}{\pi_k}\mu_k^{(k)}\right) + M^1, \quad \forall i \in \mathcal{N}_-\backslash\{k\} \tag{18b}$$

Thus, in the k^{th} subproblem, a lower bound to M^1 is

$$M^1 \geq \max_{i\in\mathcal{N}_-\backslash\{k\}} \left\{\pm 2\tilde{D}_i\left(-\mu_i^{(k)} + \left(\mu_k^{(k)}\pi_i\right)/\pi_k\right)\right\}. \tag{19}$$

For M^2, it must satisfy

$$\gamma_{ij}^{(k)} + M^2 \geq 0 \quad \forall (i,j) \in \mathcal{A} \tag{20}$$

Thus, a lower bound to M^2 is

$$M^2 \geq \max_{(i,j)\in\mathcal{A}} \left\{-\gamma_{ij}^{(k)}\right\} \quad \forall k \in \mathcal{N}_-. \tag{21}$$

Note that both lower bounds in (19) and (21) involve dual solutions to (17) and thus cannot be obtained *a priori* to solving model (17). We describe an iterative approach using (19) and (21) to compute valid M^1 and M^2 as follows. We start with sufficiently large M^1 and M^2 in (17) to obtain optimal dual solutions $\mu^{(k)}$ and $\gamma^{(k)}$, using which we update the values of M^1 and M^2 based on (19) and (21), respectively. We re-compute (17) for possibly new optimal solutions of $\mu^{(k)}$ and $\gamma^{(k)}$ using the new M^1 and M^2. We repeat the process until the improvements of M^1 and M^2 are sufficiently small. Such an iterative approach for updating big-M coefficients in MILP models has been discussed and implemented by, e.g., Qiu et al. [50], who iteratively strengthen big-M coefficients for MILP models with 0-1 Knapsack structures, to reformulate chance-constrained linear programs with finite samples. In our later computational studies the values of M^1 and M^2 do not significantly affect the solution time and we use $M^1 = M^2 = 1000$ in all our instances.

We repeatedly solve Model (17) for every $k \in \mathcal{N}_-$. Suppose that the $\hat{k}^{\text{th}}$ subproblem yields the maximum objective value among all subproblems, and solution $(\alpha_i^{(\hat{k})}, \mu_i^{(\hat{k})}, \nu_i^{(\hat{k})}, \sigma_{ij}^{(\hat{k})})$ is optimal to the original Problem (15). As a result,

$$y \geq \sum_{i \in \mathcal{N}_- \setminus \{\hat{k}\}} \alpha_i^{(\hat{k})} - \frac{\Pi}{\pi_{\hat{k}}} \hat{\mu}_{\hat{k}}^{(\hat{k})} + \sum_{i \in \mathcal{N}_+} S_i \nu_i^{(\hat{k})} + \sum_{(i,j) \in \mathcal{A}} a_{ij} \sigma_{ij}^{(\hat{k})} x_{ij}, \tag{22}$$

which we refer to as an optimality cut to be generated into set $L_2(y, \mathbf{x}) \geq 0$.

Theorem 3. Cut (22) is valid to **MP1**, for all $\mathbf{x} \in \mathcal{R}_x$.

Proof. If $\mathbf{x} \in \mathcal{R}_x$, then $V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ is feasible for any $\mathbf{I} \in U_I, \mathbf{D} \in U_D$. Recall model (15) in which variables x_{ij}, $\forall (i,j) \in \mathcal{A}$ only exist in the objective function. The right-hand side of (22) provides a feasible objective of $\max_{\mathbf{I} \in U_I, \mathbf{D} \in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ for any $\mathbf{x} \in \mathcal{R}_x$. This completes the proof. □

3.4 Cutting-Plane Algorithms

Algorithm 1 demonstrates the details of a cutting-plane algorithm for solving the RND problem in (3). At each iteration, the algorithm solves a relaxed **MP1** to obtain a solution $\hat{\mathbf{x}}$ and a lower bound of the second-stage recourse cost y, and then solves the separation problem $\text{SP}_{\mathcal{R}}^{\text{MILP}}$. If $R(\mathbf{x}) < 0$, a feasibility cut (13) is generated into $L_1(\mathbf{x}) \geq 0$, and we re-solve **MP1**. Otherwise, we calculate an upper bound of y by computing $\max_{\mathbf{I} \in U_I, \mathbf{D} \in U_D} V(\hat{\mathbf{x}}, \mathbf{D}, \mathbf{I})$. If there exists a positive gap between the current best upper and lower bounds that is greater than ϵ, we generate an optimality cut (22) into set $L_2(y, \mathbf{x}) \geq 0$, and solve **MP1** again.

Recall that in **MP1**, none of the constraints (7b)–(7d) reflect flow balance in the network, indicating that solutions given by **MP1** cannot be guaranteed to satisfy flow balance at every node. (We later in Section 6.3 demonstrate that many feasibility cuts will be added into **MP1** before very few optimality cuts are generated.) Here we consider an alternative master problem formulation, to improve the quality of $\mathbf{x}$ computed by **MP1** at the first stage.

$$\begin{aligned}
\textbf{Alt-MP1}: \quad \min \quad & \rho \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} + (1-\rho) y && \text{(23a)} \\
\text{s.t.} \quad & \text{(1b)–(1f)} \\
& y \geq \sum_{i \in \mathcal{N}_+} h_i g_i + \sum_{(i,j) \in \mathcal{A}} d_{ij} f_{ij} && \text{(23b)} \\
& L_1(\mathbf{x}) \geq 0 && \text{(23c)} \\
& L_2(y, \mathbf{x}) \geq 0 && \text{(23d)} \\
& x_{ij} \in \{0,1\} \ \forall (i,j) \in \mathcal{A}, \quad y \geq 0, && \text{(23e)}
\end{aligned}$$

where (1b)–(1f) and (23b) ensure that solution $\mathbf{x}$ given by **Alt-MP1** must satisfy flow balance at all the nodes and y equals to the corresponding $V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ for given $\mathbf{D}$ and $\mathbf{I}$. It follows that for

Algorithm 1 Cutting-plane algorithm for solving the RND problem in (3)

Input: An instance of problem (3).
Output: An ϵ-optimal solution to problem (3), or no feasible solution exits.
Step 0: Set $x_{ij} = 1,\ \forall (i,j) \in \mathcal{A}$. Solve $\mathrm{SP}_{\mathcal{R}}^{\mathrm{MILP}}$ in (12).
if $R(\mathbf{x}) = 0$ **then**
  go to **Step 1**.
else
  the instance is not feasible; **exit the algorithm.**
end if
Step 1: Solve **MP1** in (7) to obtain the current optimal solutions $\bar{\mathbf{x}}$ and $\bar{y}$.
Step 2: Fix $\mathbf{x} = \bar{\mathbf{x}}$, and solve $\mathrm{SP}_{\mathcal{R}}^{\mathrm{MILP}}$ in (12).
if $R(\mathbf{x}) = 0$ **then**
  go to **Step 3**.
else
  generate feasibility cut (13) into **MP1**, and go to **Step 1**.
end if
Step 3: Solve subproblem (17) for every $k \in \mathcal{N}_-$, and choose solution $\hat{y}$ that yields the maximum objective value of all subproblems.
if $(\hat{y} - \bar{y}) \leq \epsilon \hat{y}$ **then**
  $\bar{\mathbf{x}}$ is an ϵ-optimal and return the optimal objective $\rho \sum_{(i,j)\in\mathcal{A}} c_{ij}\bar{x}_{ij} + (1-\rho)\bar{y}$
else
  generate optimality cut (22) into **MP1**, and go to **Step 1**.
end if

any $\mathbf{I}' \in U_I, \mathbf{D}' \in U_D$,

$$\min_{\mathbf{x}\in\mathcal{R}_x} \rho \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho)V(\mathbf{x},\mathbf{D}',\mathbf{I}') \leq \min_{\mathbf{x}\in\mathcal{R}_x} \rho \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho) \max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I}). \quad (24)$$

Together with cuts in $L_1(\mathbf{x}) \geq 0$ and $L_2(y,\mathbf{x}) \geq 0$, Inequality (24) shows that the optimal objective value of **Alt-MP1** yields a lower bound to RND. Meanwhile, given feasible $\mathbf{x}$, we obtain an upper bound to the optimal objective value. At each iteration, we fix $\mathbf{I}$ and $\mathbf{D}$, which are computed based on subproblems (12) and (17). We develop an improved Algorithm 1, named "Alt-Algorithm 1" by revising Steps 0, 1, 2, and 3 in Algorithm 1. The key is to always record the optimal values of $\mathbf{D}$ and $\mathbf{I}$ after solving subproblem $\mathrm{SP}_{\mathcal{R}}^{\mathrm{MILP}}$ and then re-solve **Alt-MP1** in (23) by fixing the recorded values of $\mathbf{D}$ and $\mathbf{I}$. Appendix B presents a full description of Alt-Algorithm 1.

4 Models and Algorithms for Optimizing S-RND

In this section, we describe solution methods for S-RND in (6), which is a two-stage stochastic program with an additional robust constraint (5). We continue using the budgeted uncertainty sets U_I and U_D in (8) and (9), respectively.

4.1 Valid Feasibility and Optimality Cuts

We develop a cutting-plane algorithm for Model (6) that reuses feasibility cuts (13) and also uses generalized optimality cuts (22). The following formulation **MP2** describes the master problem of

S-RND:

$$\mathbf{MP2}: \quad \min \quad \rho \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} + (1-\rho)\sum_{\omega=1}^{N} \eta^{\omega}/N \tag{25a}$$

$$\text{s.t.} \quad L_1(\mathbf{x}) \geq 0 \tag{25b}$$

$$L_2(\eta^{\omega}, \mathbf{x}) \geq 0, \ \omega = 1, 2, \ldots, N \tag{25c}$$

$$x_{ij} \in \{0,1\}, \ \forall (i,j) \in \mathcal{A}, \ \eta^{\omega} \geq 0, \ \omega = 1, 2, \ldots, N \tag{25d}$$

where η^{ω} provides a lower bound to $V(\mathbf{x}, \mathbf{D}^{\omega}, \mathbf{I}^{\omega})$, $\omega = 1, 2, \ldots, N$ for any $\mathbf{x}$. Sets $L_1(\mathbf{x}) \geq 0$ and $L_2(\eta^{\omega}, \mathbf{x}) \geq 0$ respectively correspond to feasibility and optimality cuts to be generated from subproblems.

To derive feasibility cuts in the set $L_1(\mathbf{x}) \geq 0$ and enforce $\mathbf{x} \in \mathcal{S}_x$, first note that cut (13) is still valid to **MP2** because $\mathcal{S}_x \subseteq \mathcal{R}_x$. Also, we require $\max_{\mathbf{I}\in U_I, \mathbf{D}\in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I}) \leq \mathcal{L}$ for any $\mathbf{x} \in \mathcal{S}_x$. Through the same method of optimizing $\max_{\mathbf{I}\in U_I, \mathbf{D}\in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I})$, we cut off all solutions $\mathbf{x} \in \mathcal{R}_x$ that will make $\max_{\mathbf{I}\in U_I, \mathbf{D}\in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I}) > \mathcal{L}$. Suppose that $(\hat{k}, \hat{\alpha}_i^{(\hat{k})}, \hat{\mu}_{\hat{k}}^{(\hat{k})}, \hat{\nu}_i^{(\hat{k})}, \hat{\sigma}_{ij}^{(\hat{k})})$ is an optimal solution to $\max_{\mathbf{I}\in U_I, \mathbf{D}\in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I})$. We require

$$\mathcal{L} \geq \sum_{i\in\mathcal{N}_-\setminus\{\hat{k}\}} \hat{\alpha}_i^{(\hat{k})} - \frac{\Pi}{\pi_{\hat{k}}}\hat{\mu}_{\hat{k}}^{(\hat{k})} + \sum_{i\in\mathcal{N}_+} S_i\hat{\nu}_i^{(\hat{k})} + \sum_{(i,j)\in\mathcal{A}} a_{ij}\hat{\sigma}_{ij}^{(\hat{k})}x_{ij}. \tag{26}$$

The validity of cut (26) can be verified given that the right-hand side of (26) is feasible to $V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ for any $\mathbf{x} \in \mathcal{S}_x$, which is then bounded by $\mathcal{L}$. Cut (26) and cut (13) constitute $L_1(\mathbf{x}) \geq 0$ in **MP2**.

To derive optimality cuts in the set $L_2(\eta^{\omega}, \mathbf{x}) \geq 0$, given $\mathbf{x}$ and parameter $\mathbf{D}^{\omega}, \mathbf{I}^{\omega}$ at the ω^{th} subproblem to **MP2**, we formulate the dual of $V(\mathbf{x}, \mathbf{D}^{\omega}, \mathbf{I}^{\omega})$ as:

$$\eta^{\omega} = \quad \max \quad \left\{ \sum_{i\in\mathcal{N}_-} -D_i^{\omega}\mu_i^{\omega} + \sum_{i\in\mathcal{N}_+} S_i\nu_i^{\omega} + \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij}I_{ij}^{\omega}\gamma_{ij}^{\omega} \ : \ \text{(14b)–(14e)} \right\} \tag{27}$$

Therefore, the cuts in $L_2(\eta^{\omega}, \mathbf{x}) \geq 0$, $\omega = 1, 2, \ldots, N$ can be obtained by the standard Benders procedures given linear (14b)–(14e). Following the weak duality,

$$\eta^{\omega} \geq \sum_{i\in\mathcal{N}_-} -D_i^{\omega}\hat{\mu}_i^{\omega} + \sum_{i\in\mathcal{N}_+} S_i\hat{\nu}_i^{\omega} + \sum_{(i,j)\in\mathcal{A}} a_{ij}x_{ij}I_{ij}^{\omega}\hat{\gamma}_{ij}^{\omega} \tag{28}$$

where $\hat{\mu}_i^{\omega}, \hat{\nu}_i^{\omega}$ and $\hat{\gamma}_{ij}^{\omega}$ are optimal dual solutions to Model (27).

Here we utilize the derivation of optimality cuts in RND to build feasibility cuts in S-RND. Later in our computation, we also use the optimal objective value of RND to design the threshold value for bounding the worst-case flow cost in S-RND. Although the two models share similar solution methods, we point out that they are designed for different data availability cases – S-RND requires knowing finite samples of the uncertainties to compute the expectation-based objective value, while RND only needs demand upper and lower bounds as well as parameters Π and Γ in sets U_I and U_D. In data-scarce environment, it is more appropriate to use RND although we computationally show later that S-RND yields similar robust solutions as RND but lower cost, when information about the uncertainty become available.

4.2 Cutting-Plane Algorithm

Proposition 1 states that for sufficiently large N, the sample average function $\sum_{\omega=1}^{N} V(\mathbf{x}, \mathbf{D}^{\omega}, \mathbf{I}^{\omega})/N$ can provide a reasonable statistical estimate for $E_{\xi}[V(\mathbf{x}, \mathbf{D}_{\xi}, \mathbf{I}_{\xi})]$. Following similar setups as the

SAA method, we solve M i.i.d samples, each with N realizations of the random $(\mathbf{I}, \mathbf{D})$. Algorithm 2 demonstrates the steps of solving Model (6) for each sample m, $m = 1, \ldots, M$. For every sample m,

Algorithm 2 Cutting-plane algorithm for solving problem (6) in sample m.

Input: An instance of problem (6) with $(\mathbf{I}^{\omega,m}, \mathbf{D}^{\omega,m})$, $\omega = 1, 2, \ldots, N$ being the realizations of $(\mathbf{I}, \mathbf{D})$ in sample m.
Step 0: Set $x_{ij} = 1$, $\forall (i,j) \in \mathcal{A}$. Solve $\text{SP}_{\mathcal{R}}^{\text{MILP}}$ in (12) and subproblem (17) for every $k \in \mathcal{N}_-$.
if the optimal objective value of (12) $= 0$ and the maximum of the optimal objective value of each (17) $\leq \mathcal{L}$ **then**
  go to **Step 1**.
else
  report that no feasible solution to this instance, and exit.
end if
Step 1: Solve **MP2** in (25) to obtain solutions $\bar{\mathbf{x}}$ and $\bar{\eta}^\omega$, $\omega = 1, 2, \ldots, N$.
Step 2: Fix $\mathbf{x} = \bar{\mathbf{x}}$, and solve $\text{SP}_{\mathcal{R}}^{\text{MILP}}$ in (12).
if the optimal objective value $= 0$ **then**
  go to **Step 3**.
else
  generate feasibility cut (13) into set $L_1(\mathbf{x}) \geq 0$, and go to **Step 1**.
end if
Step 3: Solve subproblem (17) for every $k \in \mathcal{N}_-$, and choose the solution with the maximum optimal objective value $\hat{y}$.
if $\hat{y} \leq \mathcal{L}$ **then**
  go to **Step 4**.
else
  generate feasibility cut (26) into set $L_1(\mathbf{x}) \geq 0$, and go to **Step 1**.
end if
Step 4: Solve subproblem (27) for $\omega = 1, 2, \ldots, N$. Calculate $\sum_{\omega=1}^{N} \eta^\omega / N$.
if $\sum_{\omega=1}^{N} \eta^\omega / N - \sum_{\omega=1}^{N} \bar{\eta}^\omega / N > \epsilon \sum_{\omega=1}^{N} \eta^\omega / N$ **then**
  generate cut (28) into set $L_2(\eta^\omega, \mathbf{x}) \geq 0$, $\omega = 1, 2, \ldots, N$, and go to **Step 1**.
else
  return $\mathbf{x}^m = \bar{\mathbf{x}}$ and $V^m = \rho \sum_{(i,j) \in \mathcal{A}} c_{ij} \bar{x}_{ij} + (1 - \rho) \sum_{\omega=1}^{N} \bar{\eta}^\omega / N$.
end if

we obtain an ϵ-optimal solution $\mathbf{x}^m$ and its corresponding optimal objective value V^m. Moreover, $\bar{V}^* = \sum_{m=1}^{M} V^m / M$ provides a statistical estimate for a lower bound to the optimal objective of S-RND. To obtain an upper bound, we choose some $\mathbf{x}^m$ computed by Algorithm 2 for any $m = 1, \ldots, M$. We generate a reference sample with realizations $(\mathbf{I}^\omega, \mathbf{D}^\omega)$, $\omega = 1, 2, \ldots, N'$ and $N' \gg N$ for post-optimization simulation. The value

$$\hat{V} = \rho \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij}^m + (1 - \rho) \sum_{\omega=1}^{N'} V(\mathbf{x}^m, \mathbf{D}^\omega, \mathbf{I}^\omega) / N'$$

provides a statistical estimate for an upper bound to the optimal objective of S-RND because $\mathbf{x}^m \in \mathcal{S}_x$. The value $\hat{V} - \bar{V}^*$ is the optimality gap.

Remark 1. The proposed approaches for RND and S-RND can be generalized to tackle NDP with multi-commodity flows, where the flow variables, supply generations, and demand parameters are all indexed by commodity in the objective (1a) and constraints (1b)–(1f). As a result, we also index the corresponding dual variables and formulate dual constraints with respect to each commodity to proceed with the diverse cutting-plane algorithms. Note that the overall sum of flows of multiple commodities is upper bounded by $a_{ij}I_{ij}x_{ij}$ on each arc (i, j) in (1f). This prevents us solving the scenario-based subproblems in each cutting-plane scheme via further decomposition by commodity. Thus, the proposed approaches could be inefficient when computing multi-commodity problems under two uncertainties and it is desirable to develop more effective algorithms by combining the Benders decomposition (i.e., row generation) with column generation [see 11, 1] for handling the coupling capacity constraint (1f). We leave this for future research and focus on single-commodity NDP in this paper.

5 Valid Inequalities

We propose additional valid inequalities that can be integrated in **MP1**, **Alt-MP1** and **MP2** to respectively improve the performance of Algorithm 1, Alt-Algorithm 1 and Algorithm 2. Compared with the generic feasibility cuts such as (13) resulted from verifying (10) for all node subsets $\widetilde{\mathcal{N}} \subseteq \mathcal{N}$, of which the number can be exponential, the valid inequalities proposed in this section are based on graph topologies (e.g., node degrees after building arcs) and arc flow capacities required by any feasible solution to RND or to S-RND.

5.1 Degree-Based Valid Inequalities

Recall that the set $\phi^+(\mathcal{N})$ contains all the incoming arcs of the nodes in a node set $\mathcal{N}$. We also denote $\phi^-(\mathcal{N})$ as the set of outgoing arcs of the nodes in set $\mathcal{N}$. Any feasible binary solution x_{ij}, $\forall (i, j) \in \mathcal{A}$ satisfies

$$\sum_{j:(j,i)\in\mathcal{A}} x_{ji} \geq 1 \quad \forall i \in \mathcal{N}_- \tag{29a}$$

$$\sum_{(i,j)\in\phi^-(\mathcal{N}_+)} x_{ij} \geq \Gamma + 1 \tag{29b}$$

$$\sum_{(j,i)\in\phi^+(\mathcal{N}_-)} x_{ji} \geq \Gamma + 1 \tag{29c}$$

$$x_{ij} + x_{ji} \leq 1 \quad \forall (i, j) \in \mathcal{A}. \tag{29d}$$

Inequality (29a) indicates that the number of incoming arcs to each demand site is no less than one, so that we can satisfy positive demand at each node $i \in \mathcal{N}_-$. Inequalities (29b)–(29c) show that $\Gamma + 1$ sets a lower bound for the number of arcs needed to be constructed. That is, because the number of arc disruptions is up to Γ, the number of arcs constructed from set $\mathcal{N}_+$ to set $\mathcal{N}_-$ needs to be at least $\Gamma + 1$ in both RND and S-RND for worst-case flows. Assuming positive arc construction and flow costs, we build no more than one arc between any pair of nodes and thus (29d) is valid.

Similarly, consider all transmission nodes in $\mathcal{N}_=$ with zero supply/demand. Define a binary variable t_j, $\forall j \in \mathcal{N}_=$ to indicate whether node j is part of a constructed path from $\mathcal{N}_+$ to $\mathcal{N}_-$ or is not connected with any constructed arc in solution $\mathbf{x}$. For each $j \in \mathcal{N}_=$, we propose

$$x_{ij} \leq 1 - t_j,\ \forall (i, j) \in \mathcal{A}, \quad x_{ji} \leq 1 - t_j,\ \forall (j, i) \in \mathcal{A} \tag{30a}$$

$$t_j + \sum_{i:(i,j)\in\mathcal{A}} x_{ij} \geq 1,\ t_j + \sum_{i:(j,i)\in\mathcal{A}} x_{ji} \geq 1, \tag{30b}$$

where inequalities (30a) indicate that we do not construct any incoming or outgoing arcs for node j if $t_j = 1$. In such a case, inequalities (30b) are satisfied. If $t_j = 0$, because node j is a transmission node, we have to construct both incoming and outgoing arcs, enforced by inequalities (30b). The above cuts ensure that we either do not build any arcs for a transmission node or build both incoming and outgoing arcs at the same time.

5.2 Capacity-Based Valid Inequalities

Consider any feasible binary solution x_{ij}, $\forall (i,j) \in \mathcal{A}$ and valid inequalities

$$\sum_{i:(i,j)\in\mathcal{A}} a_{ij}x_{ij} \geq \bar{D}_j - \tilde{D}_j \quad \forall j \in \mathcal{N}_- \tag{31a}$$

$$\sum_{(i,j)\in\phi^-(\mathcal{N}_+)} a_{ij}x_{ij} \geq \sum_{i\in\mathcal{N}_-} (\bar{D}_i - \tilde{D}_i). \tag{31b}$$

Inequalities (31a) guarantee that the total capacities of the incoming arcs of each demand node are not fewer than the minimum possible demand. Inequality (31b) guarantees that the total outgoing capacities from all the supply nodes in $\mathcal{N}_+$ meet the minimum of the total demand generated from the nodes in $\mathcal{N}_-$.

Moreover, we build valid inequalities by using the max-flow min-cut theorem [1]. We first identify the minimum cut of graph G from node set $\mathcal{N}_+$ to node set $\mathcal{N}_-$ with all arcs in $\mathcal{A}$ being constructed. Denote $A_{\min} \subseteq \mathcal{A}$ as the minimum-cut set with any arc $(i,j) \in A_{\min}$ having node $i \in \mathcal{N}_+$ and node $j \in \mathcal{N}_-$. For any feasible solution x_{ij}, $\forall (i,j) \in \mathcal{A}$, denote $f(x)$ as the sum of the flow on arcs built from $\mathcal{N}_+$ to $\mathcal{N}_-$. The following relationship holds:

$$\sum_{j\in\mathcal{N}_-} (\bar{D}_j - \tilde{D}_j) \leq f(x) \leq \sum_{(i,j)\in\mathcal{A}:\ i\in\mathcal{N}_+,\ j\in\mathcal{N}_-} a_{ij}x_{ij} \leq \sum_{(i,j)\in A_{\min}} a_{ij}x_{ij} \tag{32}$$

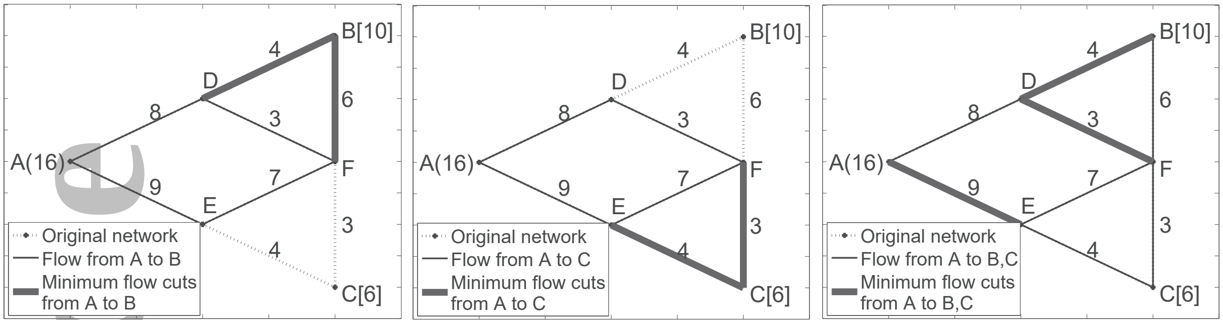
For the first inequality from left to right of (32), the feasibility of solution $\mathbf{x}$ guarantees the total amount of flow from $\mathcal{N}_+$ to $\mathcal{N}_-$ is no less than the total amount of demand on the left-hand side. This value is bounded by the total capacities of the arcs built from $\mathcal{N}_+$ to $\mathcal{N}_-$ by solution $\mathbf{x}$, and further bounded by the maximum flow (equivalent to the capacity of the minimum cut) from $\mathcal{N}_+$ to $\mathcal{N}_-$ if we construct all arcs in $\mathcal{A}$. We therefore propose the following inequality:

$$\sum_{i\in\mathcal{N}_-} (\bar{D}_i - \tilde{D}_i) \leq \sum_{(i,j)\in A_{\min}} a_{ij}x_{ij} \tag{33}$$

We develop a similar type of inequalities by deriving modified copies of graph G, denoted by G^k for each $k \in \mathcal{N}_-$, where we keep the same network structure and topologies, but modify the demand values such that we only consider a singleton demand node k for each G^k. We then find the minimum cut set from $\mathcal{N}_+$ to node k, denoted by $A^k_{\min}$, for all $k \in \mathcal{N}_-$. The following set of inequalities can be derived in light of the derivation of (33).

$$\bar{D}_k - \tilde{D}_k \leq \sum_{(i,j)\in A^k_{\min}} a_{ij}x_{ij} \quad \forall k \in \mathcal{N}_-. \tag{34}$$

An Example: Figure 1 illustrates an example where the minimum cut sets $A_{\min}$, $A^k_{\min}$, $\forall k \in \mathcal{N}_-$ are different, and so are the related valid inequalities (33), (34). Node A's supply mean value is 16, Node B and Node C's demand mean values are 10 and 6, respectively. Flow capacities are shown along the side of each arc.



(a) Subnetwork G^B with $\mathcal{N}_-^B = \{B\}$ (b) Subnetwork G^C with $\mathcal{N}_-^C = \{C\}$ (c) Network G with $\mathcal{N}_- = \{B, C\}$

Figure 1: An example for demonstrating valid inequalities (33) and (34).

In Figure 1(a) and Figure 1(b), we examine graphs G^B and G^C for nodes $B,\ C \in \mathcal{N}_-$, respectively. Their corresponding $A^B_{\min}$ and $A^C_{\min}$ are $\{(D, B), (F, B)\}$ and $\{(E, C), (F, C)\}$, respectively. We specify the corresponding inequalities (34) as

$$\bar{D}_B - \tilde{D}_B \leq 4x_{DB} + 6x_{FB}, \qquad \bar{D}_C - \tilde{D}_C \leq 4x_{EC} + 3x_{FC}.$$

For graph G with $\mathcal{N}_- = \{B, C\}$, the corresponding $A_{\min} = \{(A, E), (D, B), (D, F)\}$ highlighted in Figure 1(c). The inequality (33) is:

$$(\bar{D}_B - \tilde{D}_B) + (\bar{D}_C - \tilde{D}_C) \leq 9x_{AE} + 4x_{DB} + 3x_{DF}.$$

6 Computational Results

In this section, we demonstrate the computational efficacy of valid inequalities developed in Section 5 by applying them to randomly generated network instances. Then we implement Algorithm 1, Alt-Algorithm 1 to solve RND and use Algorithm 2 to solve S-RND based on instances generated based on real-world networks. We compute ten replications of each instance and report the average results unless specified otherwise. All the experiments are performed by using CPLEX 12.5.1 with C++ language on Workstation with Intel(R) Xeon CPU X5570 2.93GHz and 6GB memory. Note that the relaxed master problems **MP1**, **Alt-MP1**, and **MP2** are MILP models and we use branch-and-bound to optimize them in each iteration. When implementing the cutting-plane algorithms, we add feasibility/optimality cuts using `callback` functions in CPLEX for both integer and fractional temporary solutions. We generate violated cuts at each node in the branch-and-bound tree for solving the master problem as long as any exists.[1] The optimality gap tolerance is 0.01% following the default setting in CPLEX. We set the threshold for identifying violated cuts as 10^{-4} and use one computational thread.

6.1 Experimental Setup and Parameter Design

We generate test instances from the following network structures. We consider three sets of random network instances, named "RG1", "RG2", and "RG3" with 5, 6, and 7 nodes, respectively, in which

[1]Such an implementation has been discussed and shown very effective for implementing the Benders decomposition algorithm, e.g., at `http://orinanobworld.blogspot.com/2011/10/benders-decomposition-then-and-now.html`.

we randomly select pairs of nodes to form the potential arcs in $\mathcal{A}$. We also use real-world networks "ABILENE", "POLSKA", "NOBEL-US" from the online Survivable Network Design Library [46], and the Sioux-Falls road network from the Transportation Test Problems [see 10, 35]. Table 1 provides the total number of nodes in $\mathcal{N}$ and the total number of arcs in $\mathcal{A}$ in each type of random or real-world network we test.

Table 1: Graph size of each test instance

Network	RG1	RG2	RG3	ABILENE	POLSKA	NOBEL-US	Sioux-Falls
$(\lvert\mathcal{N}\rvert, \lvert\mathcal{A}\rvert)$	(5,7)	(6,16)	(7,20)	(12,30)	(12,36)	(14,42)	(24, 76)

We follow a Bernoulli trial and randomly make each node i as a supply, demand, or transmission node. The average number of supply and demand nodes is about 15%-20% of the total number of nodes in all network instances. We set the values of construction cost (c), flow cost (d), generation cost (h), arc capacity (a), and supply capacity (S) by uniformly generating integer numbers from the corresponding intervals given in Table 2.

Table 2: Parameter settings

Parameter	c	d	h	a	S
Interval	$[8, 12]$	$[1, 3]$	$[3, 5]$	$[20, 60]$	$[20, 60]$

The parameters for defining the uncertainty sets U_I and U_D represent network designers' risk preference and can be determined via, e.g., cross validation results. In this paper, for the uncertainty set U_I in (9), we consider $\Gamma = 2$ for RG1 and $\Gamma = 3$ for all the other networks; for the uncertainty set U_D in (8), we generate the values of $\bar{D}_i - \tilde{D}_i$ and $\bar{D}_i + \tilde{D}_i$ from intervals $[1, 5]$ and $[11, 15]$, respectively, for each demand node $i \in \mathcal{N}_-$. We also set $\pi_i = 1,\ \forall i \in \mathcal{N}_-$. To determine the value of Π, $D_i,\ \forall i \in \mathcal{N}_-$ are sampled uniformly between $\bar{D}_i - \tilde{D}_i$ and $\bar{D}_i + \tilde{D}_i$, and $\Pi = \max_{s\in\Omega} \sum_{i\in\mathcal{N}_-} D_i^s$ where Ω is the set of all the samples, and D_i^s presents the demand realization of node i in sample s. Table 3 provides the Π-values for all the networks. We use $M^1 = M^2 = 1000$ which are verified sufficiently large by our results.

Table 3: Values of Π for each network

Instance	RG1	RG2	RG3	ABILENE	POLSKA	NOBEL-US	Sioux-Falls
Π	8	8	15	14	7	7	10

For RND, we assume that only the parameters for defining sets U_I and U_D are given; we later generate data samples (including realizations of $\mathbf{D}$ and $\mathbf{I}$) based on the designed demand intervals, demand distribution type, and parameter Γ for S-RND and the stochastic optimization approach.

6.2 Efficacy of Valid Inequalities Added to Algorithm 1

We first test Algorithm 1 for solving RND with or without adding the valid inequalities proposed in Section 5. Tables 4 reports the results of testing RND on RG1, RG2, and RG3 instances. We compute each instance with weight parameter $\rho = 1, 0.75, 0.25$ and 0. Columns $\mathbf{t}_{\text{total}}$, $\mathbf{t}_{\text{fea}}$ and $\mathbf{t}_{\text{opt}}$ report the average CPU seconds of solving replications of each instance, of deriving each feasibility cut (13) and of deriving each optimality cut (22), respectively. Columns **Fea.Cut** and **Opt.Cut** report the number of feasibility cut (13) and optimality cut (22) being added into **MP1** when Algorithm 1 terminates. Column **Opt.Obj** reports the optimal objective value of **MP1**.

Columns **Flow.C** and **Cons.C** report the corresponding flow cost (i.e., $\max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I})$) and construction cost (i.e., $\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij}$), respectively. We report "Yes" (Y) or "No" (N) in the last two columns to indicate whether the degree-based (**DBC**) and capacity-based inequalities (**CBC**) are included in **MP1** when solving the problem.

In Table 4, adding the valid inequalities significantly reduces the number of feasibility cuts needed to ensure $\mathbf{x} \in \mathcal{R}_x$ and also reduces the CPU time. The degree-based valid inequalities (29a)–(29d), (30a)–(30b) are more effective than the capacity-based inequalities (31a)–(31b), (33), (34) in our computation, where arc capacities are generally large.

6.3 Comparison of Algorithm 1 and Alt-Algorithm 1 for RND

Recall that Alt-Algorithm 1 in Section 3.4 revises Algorithm 1 by imposing flow balance constraints at the first stage for fixed $\mathbf{I}$ and $\mathbf{D}$, aiming at quickly finding lower bounds to the optimal objective value of RND. In this section, we compare the two algorithm variants by applying them to the four types of real-world networks, ABILENE, POLSKA, NOBELUS, and Sioux-Falls. We add the valid inequalities in Section 5 into both algorithms, given their efficient performance shown over the randomly generated networks in the previous section. We report the results in Table 5.

In Table 5, for each instance using the same ρ, the CPU time taken by Alt-Algorithm 1 is significantly less than the time of Algorithm 1. The number of feasible cuts used in Alt-Algorithm 1 is also much fewer than the ones generated by Algorithm 1. As column **Fea.Cut** shows, the number of feasibility cuts is much fewer taken by Alt-Algorithm 1, which leads to time reduction and thus makes Alt-Algorithm 1 more efficient.

Moreover, for both algorithms, solving the master problem accounts for the majority of the total CPU time, while little time is spent on generating the feasibility cuts (13) and optimality cuts (22). We also add fewer optimality cuts than feasibility cuts in all instances. This is due to that **MP1** does not contain flow-balance constraints, and thus it may take many iterations to find a feasible $\mathbf{x} \in \mathcal{R}_x$ before checking the optimality conditions (see, e.g., NOBEL-US with $\rho = 0.25$).

Recall that weight parameter $\rho \in [0, 1]$ reflects the decision maker's preference on the tradeoffs between construction cost (**Cons.C**) and flow cost (**Flow.C**). As ρ decreases, the cost in **Cons.C** increases while the cost in **Flow.C** decreases. If $\rho = 0$, then we minimize $\max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I})$ without taking into account the design cost $\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij}$. The algorithm simply builds all necessary arcs at the first stage. This case can be solved very fast as shown in above tables.

6.4 Results of Algorithm 2 for S-RND

We solve S-RND using the SAA method as a common approach for solving stochastic programs with random parameters by reducing the scenario set to a manageable size (see, e.g., [33, 52]). Following common parameter settings in the SAA literature, we employ Monte Carlo sampling and generate five i.i.d. samples ($M = 5$), each having $N = 100$ or 200 scenarios.[2] We sample $I_{ij},\ \forall (i,j) \in \mathcal{A}$ as Bernoulli random variables with "success" probability $\Gamma/|\mathcal{A}|$, i.e., $I^q_{ij} = 0,\ \forall (i,j) \in \mathcal{A}$ with probability $\Gamma/|\mathcal{A}|$, and generate $D_i,\ i \in \mathcal{N}_-$ as integer values rounded from random realizations of Uniform, Normal, Triangular, and Gamma distributions, separately. We test these four commonly used types of distributions as the underlying true distributions of the random demand, to compare optimal solutions given by the S-RND and their sensitivity to the assumption of distribution type when decision makers do not have full knowledge of the true distribution. Therefore, in the following

[2] The theoretical sample size needed to ensure small optimality gap is much larger than the selected sample sizes [33]. In most of our instances, we show in Table 6 that $N = 100$ and 200 with $M = 5$ result in SAA gaps $\leq 2\%$.

Table 4: RG1, RG2, and RG3 instances solved by Algorithm 1 with or without valid inequalities

Instance	ρ	t_{total} (s)	t_{fea} (s)	t_{opt} (s)	Fea.Cut	Opt.Cut	Opt.Obj	Flow.C	Cons.C	DBC	CBC
RG1	1	0.128	0	0	0	1	50	130	50	Y	Y
	0.75	0.040	0	0	0	1	70	130	50		
	0.25	0.053	0	0	0	2	65	40	140		
	0	0.032	0	0	0	1	40	40	160		
	1	0.051	0	0	2	1	50	130	50	Y	N
	0.75	0.052	0	0	2	1	70	130	50		
	0.25	0.054	0	0	2	1	65	40	140		
	0	0.029	0	0	0	1	40	40	160		
	1	0.093	0	0	6	1	50	130	50	N	Y
	0.75	0.093	0.001	0	6	1	70	130	50		
	0.25	0.128	0.001	0	10	2	65	40	140		
	0	0.041	0	0	0	1	40	40	160		
	1	0.621	0.010	0	57	1	50	130	50	N	N
	0.75	0.508	0.004	0	57	1	70	130	50		
	0.25	0.927	0.001	0	103	1	65	40	140		
	0	0.131	0.001	0	6	2	40	40	160		
RG2	1	0.078	0	0	0	1	30	84	30	Y	Y
	0.75	0.114	0	0	0	5	41.25	72	31		
	0.25	0.322	0	0	0	13	61.75	72	31		
	0	0.054	0	0	0	2	72	72	86		
	1	0.055	0	0	3	1	30	84	30	Y	N
	0.75	0.173	0	0	8	5	41.25	72	31		
	0.25	0.161	0	0.001	8	5	61.75	72	31		
	0	0.147	0	0	2	4	72	72	114		
	1	0.193	0	0	5	1	30	84	30	N	Y
	0.75	0.972	0	0	38	5	41.25	72	31		
	0.25	1.501	0.002	0	68	10	61.75	72	31		
	0	0.129	0	0	3	2	72	72	129		
	1	18.652	0.022	0	463	1	30	84	30	N	N
	0.75	73.249	0.028	0	906	5	41.25	72	31		
	0.25	69.437	0.048	0	915	5	61.75	72	31		
	0	0.134	0.001	0	9	2	72	72	157		
RG3	1	9.708	0.010	0	166	1	41.00	156	41	Y	Y
	0.75	26.499	0.029	0.001	419	5	69.75	156	41		
	0.25	21.804	0.014	0	408	4	127.25	156	41		
	0	0.133	0	0	6	2	156.00	156	125		
	1	10.103	0.014	0	293	1	41.00	156	41	Y	N
	0.75	35.406	0.027	0	607	4	69.75	156	41		
	0.25	32.791	0.017	0.001	579	8	127.25	156	41		
	0	0.138	0.001	0	7	3	156.00	156	125		
	1	15.302	0.021	0	375	1	41.00	156	41	N	Y
	0.75	52.588	0.047	0	911	7	69.75	156	41		
	0.25	64.361	0.035	0.001	708	5	127.25	156	41		
	0	0.137	0.001	0	7	2	156.00	156	125		
	1	11.096	0.036	0	361	2	41.00	156	41	N	N
	0.75	78.427	0.072	0.001	1018	8	69.75	156	41		
	0.25	70.606	0.045	0.001	829	6	127.25	156	41		
	0	0.139	0.001	0	11	4	156.00	156	125		

Table 5: Results of Algorithm 1 and Alt-Algorithm 1 for solving RND on ABILENE, POLSKA, NOBEL-US, and Sioux-Falls networks with inequalities in Section 5 added to **MP1**

Algorithm	Instance	ρ	t_{total} (s)	t_{fea} (s)	t_{opt} (s)	Fea.Cut	Opt.Cut	Opt.Obj	Flow.C	Cons.C
Algorithm 1	ABILENE	1	2.102	0.004	0	49	1	38.00	151	38
		0.75	424.925	0.100	0.002	1509	24	66.25	151	38
		0.25	1139.281	0.196	0.009	2470	128	111.25	132	49
		0	7.032	0.003	0.003	42	55	132.00	132	133
	POLSKA	1	39.372	0.019	0	242	1	70.00	270	70
		0.75	393.281	0.072	0.003	1164	49	111.75	228	73
		0.25	176.093	0.042	0.002	745	32	189.25	228	73
		0	2.606	0.007	0	75	22	228.00	228	122
	NOBEL-US	1	7.262	0.003	0	84	1	54.00	181	54
		0.75	497.081	0.078	0.001	1211	20	85.75	181	54
		0.25	1496.347	0.240	0.005	3037	53	139.00	170	143
		0	12.349	0.013	0.009	138	81	162.00	162	185
	Sioux-Falls	1	11.170	0.007	0	112	1	62.00	219	62
		0.75	387.109	0.096	0.002	1038	34	101.25	219	62
		0.25	124.863	0.251	0.007	2691	61	171.25	183	136
		0	5.791	0.016	0.006	153	70	177.00	177	192
Alt-Algorithm 1	ABILENE	1	1.766	0	0	38	1	38.00	151	38
		0.75	6.121	0.011	0	81	1	66.25	151	38
		0.25	2.358	0.003	0	50	1	111.25	132	49
		0	1.147	0	0.002	0	25	132.00	132	133
	POLSKA	1	34.375	0.019	0	237	6	70.00	270	70
		0.75	29.286	0.018	0.002	207	22	111.75	228	73
		0.25	329.303	0.075	0.056	861	724	189.25	228	73
		0	3.593	0	0	2	4	228.00	228	122
	NOBEL-US	1	12.094	0.012	0.001	84	5	54.00	181	54
		0.75	10.277	0.008	0	80	8	85.75	181	54
		0.25	45.625	0.024	0.014	227	97	139.00	170	143
		0	22.936	0.007	0.010	58	130	162.00	162	185
	Sioux-Falls	1	7.334	0.014	0.001	112	4	62.00	219	62
		0.75	17.268	0.015	0	98	13	101.25	219	62
		0.25	45.682	0.043	0.027	261	79	171.25	183	136
		0	14.005	0.004	0.008	17	110	177.00	177	192

we match the parameters used by the four types of distributions, so that they have the same mean and support, but differ in distribution type.

For Uniform distribution, we generate $D_i, i \in \mathcal{N}_-$ over the interval $[\bar{D}_i - \tilde{D}_i, \bar{D}_i + \tilde{D}_i]$. Also, to link the scenario-based model with the previous RND model, we set the parameters describing the four distributions according to parameters describing the uncertainty sets U_D. For Normal distribution $\mathcal{N}(\mu, \sigma^2)$, we generate $D_i, i \in \mathcal{N}_-$ by choosing $\mu = \bar{D}_i$ and $\sigma = \tilde{D}_i$. The probability density function of the Triangular distribution is given by

$$f(x) = \begin{cases} 0 & \text{for } x < a, \\ \frac{2(x-a)}{(b-a)(c-a)} & \text{for } a \le x < c, \\ \frac{2}{b-a} & \text{for } x = c, \\ \frac{2(b-x)}{(b-a)(b-c)} & \text{for } c < x \le b, \\ 0 & \text{for } b < x, \end{cases}$$

for which we set $a = \max\{0, \bar{D}_i - \sqrt{6}\tilde{D}_i\}$, $b = \bar{D}_i + \sqrt{6}\tilde{D}_i$ and $c = \bar{D}_i$. For the Gamma distribution, we set its mean as $\bar{D}_i$ and mode as $\tilde{D}_i$.

Given the results of $\rho = 0, 0.25, 0.75, 1$ in Table 5, we test the S-RND model with $\rho = 0.25$ and 0.75 by using Algorithm 2 and report the results in Table 6. Note that for each S-RND instance, $V^* = \min_{\mathbf{x} \in \mathcal{R}_x} \max_{\mathbf{I} \in U_I, \mathbf{D} \in U_D} V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ equals to the optimal objective value of the RND with $\rho = 0$ in its formulation. Correspondingly, we set the robust parameter $\mathcal{L}$ in S-RND as 110% and 120% of V^*, indicating that the worst-case performance of the optimal S-RND solution is not more than 1.1 or 1.2 times V^* that is the smallest worst-case $V(\mathbf{x}, \mathbf{D}, \mathbf{I})$ possibly achieved by any solution $\mathbf{x}$.

In Table 6, columns $\mathbf{t}_{\max}, \mathbf{t}_{\min}$ and $\mathbf{t}_{\text{avg}}$ report the maximum, minimum and average CPU time (seconds) for solving replications of each instance. Columns **Fea.Cut**, **$\mathcal{L}$.Cut** and **B.Cut** respectively report the number of feasibility cuts (13), (26) and the optimality cut (28) being added into **MP2** when Algorithm 2 terminates. Columns **LB** and **UB** provide the sampling-based lower and upper bounds of the optimal objective value; column **Gap** reports the gaps between the two bounds.

Based on Table 6, the average CPU time for solving each instance slowly increases when N increases from 100 to 200. The most important factor affecting the CPU time is the number of feasibility cuts, which does not depend on the choice of N. In contrast, the number of optimality cuts, which is directly related to N, increases almost linearly on N. It is not surprising that the optimal objective values of S-RND are much smaller than those of RND – the latter considers the worst-case cost rather than the expected cost as in S-RND, with respect to the same uncertainty sets U_I and U_D. Also note that the gaps between the upper and lower bounds of the optimal objective values are all within 2% for both $N = 100$ and $N = 200$, indicating that the two selected sample sizes with $M = 5$ are appropriate for conducting the SAA approach.

According to Table 6, changing the robust parameter $\mathcal{L}$ has little impact on the optimal objective value. Increasing $\mathcal{L}$ significantly increases CPU time since the robust constraint in (5) becomes less restrictive. In fact, when $\mathcal{L} = 120\% V^*$ we have the robust requirement in (5) completely relaxed in all the instances we tested (i.e., $\mathcal{S}_x = \mathcal{R}_x$). Therefore, the solution time for all instances with $\mathcal{L} = 120\% V^*$ in Table 6 are the time for solving the SAA-based stochastic optimization model with the same sample size N. Note that they are at least twice or more of the time for solving the same instances $\mathcal{L} = 110\% V^*$. We also compute the stochastic optimization models with larger sample sizes and when $N = 500$ all the instances we tested cannot be optimized within an hour time limit. Given the small optimality gaps already archived by $N = 100$ or $N = 200$ using S-RND, we conclude that S-RND can provide cost-effective and computationally efficient solutions

Table 6: Results of Algorithm 2 for solving S-RND ($M = 5$ samples)

N	$\mathcal{L}$	Instances	Distribution	$t_{\max}$ (s)	$t_{\min}$ (s)	t_{avg} (s)	Fea.Cut	$\mathcal{L}$.Cut	B.Cut	LB	UB	Gap
100	110%V*	ABILENE	Uniform	5.36	1.48	3.77	56	13	2	67.86	69.10	1.82%
			Normal	5.39	1.56	3.54	61	11	2	67.18	67.10	-0.13%
		POLSKA	Uniform	659.22	98.04	407.21	802	9	534	102.20	103.28	1.06%
			Normal	835.18	114.76	369.87	789	8	520	102.20	103.83	1.60%
		NOBEL-US	Uniform	716.20	142.68	375.37	230	16	104	77.84	79.27	1.84%
			Normal	757.08	139.28	473.95	241	14	98	79.40	79.00	-0.50%
		Sioux-Falls	Uniform	165.38	38.26	96.03	265	6	72	101.04	100.64	-0.39%
			Normal	162.43	27.05	89.62	278	8	74	98.01	98.51	0.51%
	120%V*	ABILENE	Uniform	9.45	2.01	4.98	56	13	2	67.86	69.16	1.91%
			Normal	11.62	2.06	6.14	61	11	2	67.18	67.84	0.97%
		POLSKA	Uniform	1067.44	110.29	651.21	802	9	534	102.20	103.47	1.25%
			Normal	1207.81	153.28	607.33	789	8	520	102.20	104.23	1.99%
		NOBEL-US	Uniform	1217.42	193.05	590.16	230	16	104	77.84	78.67	1.06%
			Normal	1178.26	178.46	521.11	241	14	98	79.40	79.54	0.18%
		Sioux-Falls	Uniform	301.41	29.15	120.01	265	6	72	101.04	103.03	1.97%
			Normal	315.63	41.98	115.26	278	8	74	98.01	98.34	0.34%
200	110%V*	ABILENE	Uniform	9.38	2.22	5.84	52	12	4	65.15	66.13	1.50%
			Normal	9.38	2.22	5.84	57	10	4	65.17	66.06	1.37%
		POLSKA	Uniform	1153.64	147.06	631.18	783	10	1068	100.15	99.94	-0.21%
			Normal	1161.45	152.14	573.30	779	8	1040	99.13	100.73	1.61%
		NOBEL-US	Uniform	995.63	214.02	581.82	221	18	208	74.73	74.45	-0.37%
			Normal	1224.89	208.92	634.62	214	14	196	77.01	77.98	1.25%
		Sioux-Falls	Uniform	249.37	57.39	148.85	250	6	144	97.00	97.32	0.33%
			Normal	284.25	40.58	138.91	262	3	148	96.05	96.08	0.03%
	120%V*	ABILENE	Uniform	16.54	3.02	7.72	52	12	4	65.15	65.07	-0.12%
			Normal	12.34	3.09	7.52	57	10	4	65.17	65.74	0.88%
		POLSKA	Uniform	1668.02	165.44	1009.38	783	10	1068	100.15	102.00	1.85%
			Normal	1531.48	169.92	941.36	779	8	1040	100.15	101.56	1.41%
		NOBEL-US	Uniform	1728.06	289.58	914.75	221	18	208	74.73	75.34	0.82%
			Normal	1862.00	267.69	807.72	214	14	196	77.81	77.50	-0.40%
		Sioux-Falls	Uniform	427.46	43.73	186.02	250	6	144	97.00	97.00	0.00%
			Normal	452.35	62.97	178.65	262	3	148	96.05	96.97	0.96%

$V^* := \min_{\mathbf{x}\in\mathcal{R}_x} \max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I})$ computed from RND with $\rho = 0$.

with robust guarantee of lowering $\max_{\mathbf{I}\in U_I,\mathbf{D}\in U_D} V(\mathbf{x},\mathbf{D},\mathbf{I})$, as compared to the standard stochastic optimization approach.

We further test Algorithm 2 for optimizing the random networks based on more distribution types (i.e., Uniform, Triangular, Normal, and Gamma) for generating the i.i.d. samples. Figure 2 illustrates how the optimality gaps of RG1 instances change according to (i) different distributions of the demand, (ii) parameter $\mathcal{L}$, and (iii) sample size N. In general, the gaps decrease as the sample size N increases and they are all relatively small for any parameter we test. The results in RG2 and RG3 are similar to the one of RG1 shown in Figure 2, and are omitted here.

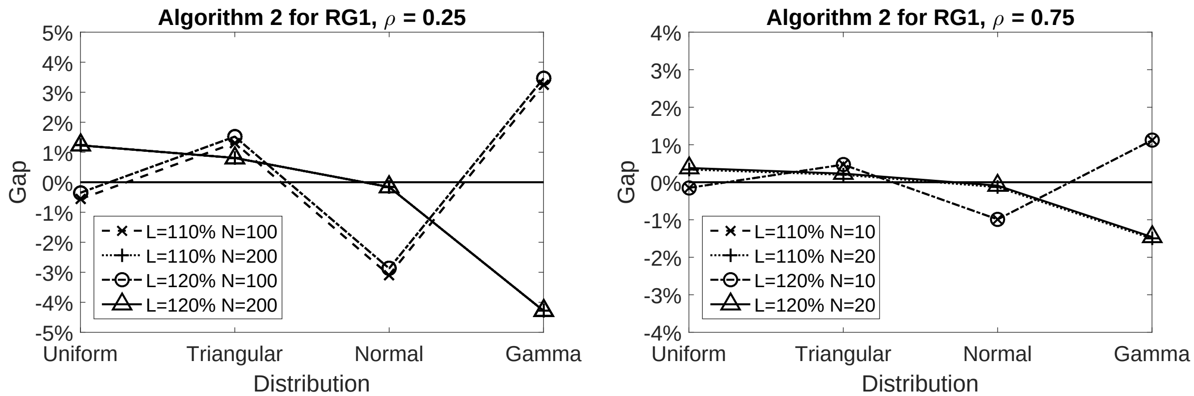


Figure 2: SAA gaps in RG1 instances provided by different demand distributions, robust parameter $\mathcal{L}$ and sample size N for $\rho = 0.25$ and $\rho = 0.75$

Figure 3 illustrates the changes of optimality gaps in networks including RG1, RG2, RG3, and ABILENE, when $N = 100$, given different values of parameter $\mathcal{L}$, and distribution types of the demand. In general, the gaps are small even we generate the samples from two very different distributions (i.e., Uniform and Normal), for both $\mathcal{L}$-values we tested. The results for $N = 200$ are quite similar and thus are omitted. This indicates that the results of S-RND are insensitive to the assumption of distribution type imposed to the same data, and the approach can be applied to cases when the distributional information is ambiguously known.

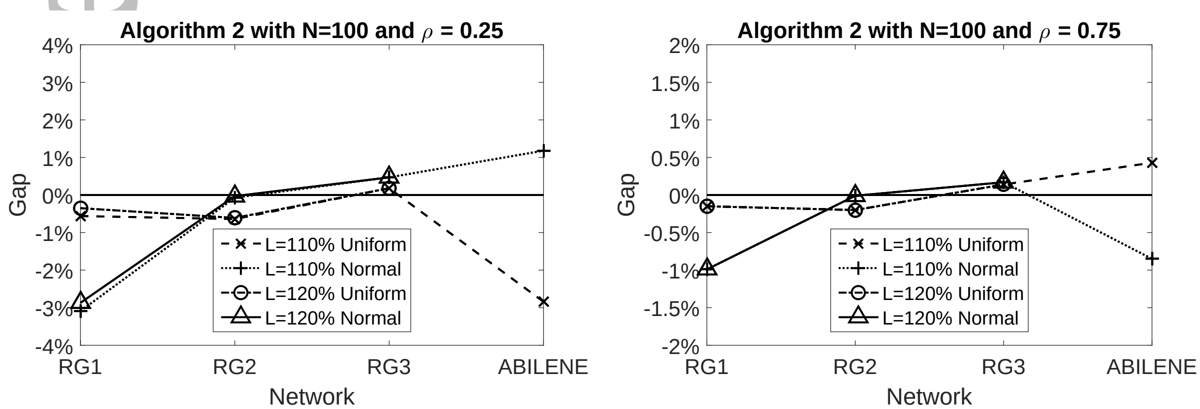


Figure 3: SAA gaps in RG1, RG2, RG3 and ABILENE instances under different demand distributions and parameter $\mathcal{L}$ when $N = 100$ for $\rho = 0.25$ and $\rho = 0.75$

6.5 Computational Result Summary

We summarize the key observations from the above computational results as follows.

1. The valid inequalities developed in Section 5 significantly shorten the CPU time and reduce the number of feasibility cuts generated for optimizing RND. Alt-Algorithm 1 is much faster than Algorithm 1 by generating even fewer number of feasibility cuts, when both are implemented with the valid inequalities.

2. To improve the computational efficiency of Algorithm 2 for S-RND, we can use small sizes of data samples to derive results that are less conservative compared with the ones yielded by RND, while very little additional effort is added to the computation of a standard stochastic programming model that requires full knowledge of the distributional information and large-scale samples. The choice of the robust parameter $\mathcal{L}$ that bounds the worst-case recourse cost does not significantly impact the outcomes of the S-RND model. Thus, one can use S-RND to easily trade off between the computational effort and the result quality solution performance for stochastic network design under demand and topological uncertainties.

3. S-RND yields lower cost of network design and operations as compared to RND, when more information of the uncertainties are available for generating discrete samples used by S-RND. When only the bounds of the uncertainties are known, RND provides conservative and robust solutions to guarantee high performance under extreme cases when both S-RND and the stochastic optimization approach are not applicable.

4. Our results also show that the Algorithm 2 can work with different type of distributions, while yielding similar objective values for each type. Algorithm 2 shows its potential to be implemented as a data-driven method and to deal with unknown distributional information of the two sources of uncertainties analyzed in this paper.

7 Conclusions

In this paper, we incorporated two sources of uncertainties (i.e., demand and topological randomness) into stochastic network design. We developed cutting-plane algorithms for solving the problem modeled via two different approaches. The first approach (i.e., RND) used robust optimization by assuming budgeted uncertainty sets, for which we developed both feasibility and optimality cuts based on duality theorems and exact linearization approaches. In the second approach (S-RND), we assumed limited data and formulated a two-stage stochastic programming model based on finite realizations of the two uncertainties, in which we imposed a constraint to bound the worst-case cost of flow and supply generation. The two approaches yielded intractable MILP models and via extensive computational studies, we demonstrated the computational efficacy of different algorithms and valid inequalities. Moreover, we showed that with limited data, the S-RND approach can yield much more robust solutions than solving a stochastic programming model based on the same sample set of uncertainties, while computing S-RND only adds very incremental computational effort.

Our future research includes implementing our approaches to design networks for various applications with more complex constraints in addition to flow balance constraints. We are also interested in investigating robust or semi-robust multi-commodity NDP and other data-driven stochastic optimization approaches for balancing cost and risk in related problems.

Acknowledgement

The authors sincerely thank the editors and three anonymous reviewers for their helpful comments. Siqian Shen acknowledges partial support by the National Science Foundation under grant CMMI-1433066 and the US Department of Army, grant W911NF-17-1-0102.

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, 1993.

[2] A. Altin, H. Yaman, and M. Ç. Pinar. The robust network loading problem under hose demand uncertainty: Formulation, polyhedral analysis, and computations. *INFORMS Journal on Computing*, 23(1):75–89, 2011.

[3] E. Álvarez-Miranda, V. Cacchiani, A. Lodi, T. Parriani, and D. R. Schmidt. Single-commodity robust network design problem: Complexity, instances and heuristic solutions. *European Journal of Operational Research*, 238(3):711–723, 2014.

[4] R. Andrade, A. Lisser, and N. Maculan. Multi-service multi-facility network design under uncertainty. *Annals of Operations Research*, 199(1):157–178, 2012.

[5] A. Atamtürk. On capacitated network design cut–set polyhedra. *Mathematical Programming*, 92(3):425–437, 2002.

[6] A. Atamtürk and D. Rajan. On splittable and unsplittable flow capacitated network design arc–set polyhedra. *Mathematical Programming*, 92(2):315–333, 2002.

[7] A. Atamtürk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.

[8] J. Ayoub and M. Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13(2):219–239, 2016.

[9] R. Bai, S. W. Wallace, J. Li, and A. Y.-L. Chong. Stochastic service network design with rerouting. *Transportation Research Part B: Methodological*, 60:50–65, 2014.

[10] H. Bar-Gera. Transportation test problems. `http://www.bgu.ac.il/`, April 2009.

[11] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3): 316–329, 1998.

[12] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

[13] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Belmont, MA, 1997.

[14] A. Billionnet, M.-C. Costa, and P.-L. Poirion. 2-stage robust MILP with continuous recourse variables. *Discrete Applied Mathematics*, 170:21–32, 2014.

[15] J. R. Birge and F. V. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, NY, 1997.

[16] Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing*, 25(1):13–26, 2013.

[17] V. Cacchiani, M. Jünger, F. Liers, A. Lodi, and D. R. Schmidt. Single-commodity robust network design with finite and hose demand sets. *Mathematical Programming*, 157(1):297–342, 2016.

[18] A. Chen, J. Kim, S. Lee, and Y. Kim. Stochastic multi-objective models for network design problem. *Expert Systems with Applications*, 37(2):1608–1619, 2010.

[19] T. G. Crainic, X. Fu, M. Gendreau, W. Rei, and S. W. Wallace. Progressive hedging-based metaheuristics for stochastic network design. *Networks*, 58(2):114–124, 2011.

[20] T. G. Crainic, M. Hewitt, and W. Rei. Scenario grouping in a progressive hedging-based meta-heuristic for stochastic network design. *Computers & Operations Research*, 43:90–99, 2014.

[21] T. Cui, Y. Ouyang, and Z.-J. M. Shen. Reliable facility location design under the risk of disruptions. *Operations Research*, 58(4-Part-1):998–1011, 2010.

[22] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. *INFORMS Journal on Computing*, 10(1):1–11, 1998.

[23] M. S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley & Sons, 2011.

[24] B. Fortz and M. Poss. An improved Benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, 37(5):359–364, 2009.

[25] A. Gaivoronski, G. M. Sechi, and P. Zuddas. Cost/risk balanced management of scarce resources using stochastic programming. *European Journal of Operational Research*, 216(1): 214–224, 2012.

[26] A. A. Gaivoronski. Stochastic optimization in telecommunications. In *Handbook of Optimization in Telecommunications*, pages 761–799. Springer, 2006.

[27] D. Gale. A theorem on flows in networks. *Pacific Journal of Mathematics*, 7(2):1073–1082, 1957.

[28] M. Garg and J. C. Smith. Models and algorithms for the design of survivable multicommodity flow networks with general failure scenarios. *Omega*, 36(6):1057–1071, 2008.

[29] B. Gendron, T. G. Crainic, and A. Frangioni. *Multicommodity Cpacitated Network Design*. Springer, 1999.

[30] A. J. Hoffman. Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. In *Proceedings of Symposium of Applied Mathematics*, volume 10, pages 113–127. World Scientific, 1960.

[31] K. Høyland, M. Kaut, and S. W. Wallace. A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, 24(2-3):169–185, 2003.

[32] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1): 1–21, 2005.

[33] A. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[34] A. M. Koster, M. Kutschka, and C. Raack. Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149, 2013.

[35] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975.

[36] C. Lee, K. Lee, and S. Park. Benders decomposition approach for the robust network design problem with flow bifurcations. *Networks*, 62(1):1–16, 2013.

[37] A.-G. Lium, T. G. Crainic, and S. W. Wallace. A study of demand stochasticity in service network design. *Transportation Science*, 43(2):144–157, 2009.

[38] K. L. Maass, M. S. Daskin, and S. Shen. Mitigating hard capacity constraints with inventory in facility location modeling. *IIE Transactions*, 48(2):120–133, 2016.

[39] T. Magnanti and R. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[40] S. Mattia. The robust network loading problem with dynamic routing. *Computational Optimization and Applications*, 54(3):619–643, 2013.

[41] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[42] M. Minoux. Robust network optimization under polyhedral demand uncertainty is NP-hard. *Discrete Applied Mathematics*, 158(5):597–603, 2010.

[43] S. Mudchanatongsuk, F. Ordóñez, and J. Liu. Robust solutions for network design under transportation cost and demand uncertainty. *Journal of the Operational Research Society*, 59 (5):652–662, 2007.

[44] A. G. Nahapetyan. Bilinear programming. In *Encyclopedia of Optimization*, pages 279–282. Springer, 2009.

[45] M. Ng and S. T. Waller. Reliable system-optimal network design. *Transportation Research Record: Journal of the Transportation Research Board*, 2090(1):68–74, 2009.

[46] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. Sndlib 1.0—survivable network design library. *Networks*, 55(3):276–286, 2010.

[47] G. R. Patil and S. V. Ukkusuri. System-optimal stochastic transportation network design. *Transportation Research Record: Journal of the Transportation Research Board*, 2029(-1):80–86, 2007.

[48] M. Poss. Models and algorithms for network design problems. *4OR: A Quarterly Journal of Operations Research*, 10(2):215–216, 2012.

[49] M. Poss and C. Raack. Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2):180–198, 2013.

[50] F. Qiu, S. Ahmed, S. S. Dey, and L. A. Wolsey. Covering linear programming with violations. *INFORMS Journal on Computing*, 26(3):531–546, 2014.

[51] S. Sen, R. D. Doverspike, and S. Cosares. Network planning with random demand. *Telecommunication Systems*, 3(1):11–30, 1994.

[52] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*, volume 9. SIAM, Philadelphia, PA, 2009.

[53] S. Shen and Z. Chen. Optimization models for differentiating quality of service levels in probabilistic network capacity design problems. *Transportation Research Part B: Methodological*, 58(1):71–91, 2013.

[54] J. Shu, C.-P. Teo, and Z.-J. M. Shen. Stochastic transportation-inventory network design problem. *Operations Research*, 53(1):48–60, 2005.

[55] L. V. Snyder and M. S. Daskin. Reliability models for facility location: the expected failure cost case. *Transportation Science*, 39(3):400–416, 2005.

[56] W. Y. Szeto, Y. Jiang, D. Wang, and A. Sumalee. A sustainable road network design problem with land use transportation interaction over time. *Networks and Spatial Economics*, pages 1–32, 2014.

[57] R. Taktak. *Survivability in Multilayer Networks: Models and Polyhedra*. PhD thesis, Université Paris Dauphine-Paris IX, 2013.

[58] A. Tomaszewski, M. Pióro, and M. Żotkiewicz. On the complexity of resilient network design. *Networks*, 55(2):108–118, 2010.

[59] S. V. Ukkusuri, T. V. Mathew, and S. T. Waller. Robust transportation network design under demand uncertainty. *Computer-Aided Civil and Infrastructure Engineering*, 22(1):6–18, 2007.

[60] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

APPENDIX

A Proof of Theorem 1

Proof. Given any $x_{ij} \in \{0,1\}$, $\forall (i,j) \in \mathcal{A}$, D_i, $\forall i \in \mathcal{N}_-$, and I_{ij}, $\forall (i,j) \in \mathcal{A}$, we have solution $\mathbf{x} \in \mathcal{R}_x$ if and only if constraints (1b)–(1f) are feasible. We reformulate constraints (1b)–(1f) by

$$\sum_{j:(i,j)\in\mathcal{A}} f_{ij} - \sum_{j:(j,i)\in\mathcal{A}} f_{ji} \le S_i \quad \forall i \in \mathcal{N}_+ \tag{A-1a}$$

$$-\sum_{j:(i,j)\in\mathcal{A}} f_{ij} + \sum_{j:(j,i)\in\mathcal{A}} f_{ji} \le 0 \quad \forall i \in \mathcal{N}_+ \tag{A-1b}$$

$$\sum_{j:(i,j)\in\mathcal{A}} f_{ij} - \sum_{j:(j,i)\in\mathcal{A}} f_{ji} \le -D_i \quad \forall i \in \mathcal{N}_- \tag{A-1c}$$

$$\sum_{j:(i,j)\in\mathcal{A}} f_{ij} - \sum_{j:(j,i)\in\mathcal{A}} f_{ji} \le 0 \quad \forall i \in \mathcal{N}_= \tag{A-1d}$$

$$-\sum_{j:(i,j)\in\mathcal{A}} f_{ij} + \sum_{j:(j,i)\in\mathcal{A}} f_{ji} \le 0 \quad \forall i \in \mathcal{N}_= \tag{A-1e}$$

$$f_{ij} \le a_{ij} I_{ij} x_{ij} \quad \forall (i,j) \in \mathcal{A}, \tag{A-1f}$$

where (A-1a) and (A-1b) combine the constraints (1b) and (1e). Constraints (A-1c) are from changing the equalities (1c) into inequality constraints. Constraints (A-1d) and (A-1e) are from replacing the equalities (1d) with two inequalities.

Therefore, $\mathbf{x} \in \mathcal{R}_x$ if and only if there exist $f_{ij} \ge 0,\ (i,j) \in \mathcal{A}$ such that f_{ij} satisfies Constraints (A-1a)–(A-1f). By applying the Farkas Lemma [e.g., 13], solution $\mathbf{x} \in \mathcal{R}_x$ if and only if

$$\sum_{i\in\mathcal{N}_+} S_i v_i - \sum_{i\in\mathcal{N}_-} D_i v_i + \sum_{(i,j)\in\mathcal{A}} a_{ij} I_{ij} x_{ij} w_{ij} \ge 0. \tag{A-2}$$

This result holds for all $v_i, u_i, w_{ij} \in \mathbb{R}^+$ that satisfy

$$v_i - v_j + w_{ij} - u_i + u_j \ge 0, \quad (i,j) \in \mathcal{A},\ i \in \mathcal{N}_+ \cup \mathcal{N}_=,\ j \in \mathcal{N}_+ \cup \mathcal{N}_= \tag{A-3a}$$

$$v_i - v_j + w_{ij} - u_i \ge 0, \quad (i,j) \in \mathcal{A},\ i \in \mathcal{N}_+ \cup \mathcal{N}_=,\ j \in \mathcal{N}_- \tag{A-3b}$$

$$v_i - v_j + w_{ij} + u_j \ge 0, \quad (i,j) \in \mathcal{A},\ i \in \mathcal{N}-,\ j \in \mathcal{N}_+ \cup \mathcal{N}_= \tag{A-3c}$$

$$v_i - v_j + w_{ij} \ge 0, \quad (i,j) \in \mathcal{A},\ i \in \mathcal{N}_-,\ j \in \mathcal{N}_- \tag{A-3d}$$

where variables $v_i,\ i \in \mathcal{N}$ correspond to (A-1a), (A-1c) and (A-1d), $u_i,\ i \in \mathcal{N}_+ \cup \mathcal{N}_=$ correspond to (A-1b) and (A-1e), and $w_{ij},\ (i,j) \in \mathcal{A}$ correspond to (A-1f).

Let $\mathcal{P}$ be the polyhedron described by (A-3a)–(A-3d). Note that variables u do not appear in (A-2). Thus, for (A-2), we consider the projection of $\mathcal{P}$ on v, w, which is a polyhedron denoted as $\text{Proj}_{v,w}\mathcal{P}$. Then v_i, w_{ij} belong to $\text{Proj}_{v,w}\mathcal{P}$ if and only if there exist solutions $u_i \ge 0,\ i \in \mathcal{N}_+ \cup \mathcal{N}_=$ such that (A-3a)–(A-3d) hold. Rewrite (A-3a)–(A-3c) as

$$u_i - u_j \le v_i - v_j + w_{ij}, \quad (i,j) \in \mathcal{A},\ i \in \mathcal{N}_+ \cup \mathcal{N}_=,\ j \in \mathcal{N}_+ \cup \mathcal{N}_= \tag{A-4a}$$

$$u_i \le v_i - v_j + w_{ij}, \quad (i,j) \in \mathcal{A},\ i \in \mathcal{N}_+ \cup \mathcal{N}_=,\ j \in \mathcal{N}_- \tag{A-4b}$$

$$-u_j \le v_i - v_j + w_{ij}, \quad (i,j) \in \mathcal{A},\ i \in \mathcal{N}-,\ j \in \mathcal{N}_+ \cup \mathcal{N}_= \tag{A-4c}$$

Given any v_i, w_{ij}, (A-4a)–(A-4c) hold for $u_i \ge 0$ if and only if all extreme rays $\mathbf{r} = (r_1, r_2, \ldots)^T$ of the feasible region satisfy

$$\mathbf{r} \ge 0,\ \mathbf{r} \ne 0,\ \text{and} \sum_{(i,j)\in\mathcal{A}} r_i(v_i - v_j + w_{ij}) \ge 0 \quad \forall i \notin \mathcal{N}_-,\ \text{or}\ j \notin \mathcal{N}_-. \tag{A-5}$$

Therefore,

$$\text{Proj}_{v,w}\mathcal{P} = \Big\{ v, w \ge 0 \mid v_i - v_j + w_{ij} \ge 0,\ \forall (i,j) \in \mathcal{A} \Big\}. \tag{A-6}$$

Note that (A-6) defines the dual feasible region of a network flow problem formulated on the induced subgraph $G(\mathcal{N}, \mathcal{A})$ at the second stage. Together with (A-2) and a node subset $\widetilde{\mathcal{N}} \subseteq \mathcal{N}$, it suffices to consider v_j equals 1 if $j \in \widetilde{\mathcal{N}}$ and 0 otherwise; w_{ij} equals 1 if $(i,j) \in \phi^+(\widetilde{\mathcal{N}})$ and 0 otherwise. Therefore, for any $\mathbf{D} \in U_D, \mathbf{I} \in U_I$, solution $\mathbf{x} \in \mathcal{R}_x$ if and only if

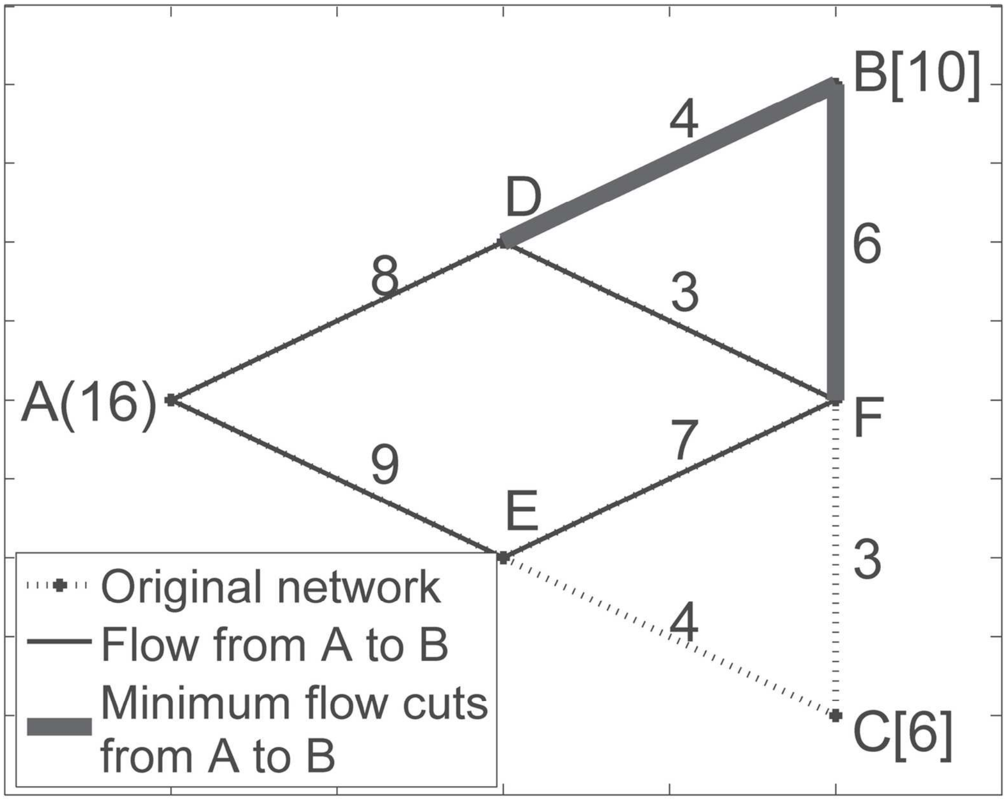
$$\sum_{i\in\mathcal{N}_+\cap\widetilde{\mathcal{N}}} S_i - \sum_{i\in\mathcal{N}_-\cap\widetilde{\mathcal{N}}} D_i + \sum_{(i,j)\in\mathcal{A}^-\cap\phi^+(\widetilde{\mathcal{N}})} a_{ij} I_{ij} x_{ij} \ge 0 \qquad \text{for all } \widetilde{\mathcal{N}} \subseteq \mathcal{N}. \tag{A-7}$$

This completes the proof. □

B Details of Alt-Algorithm 1

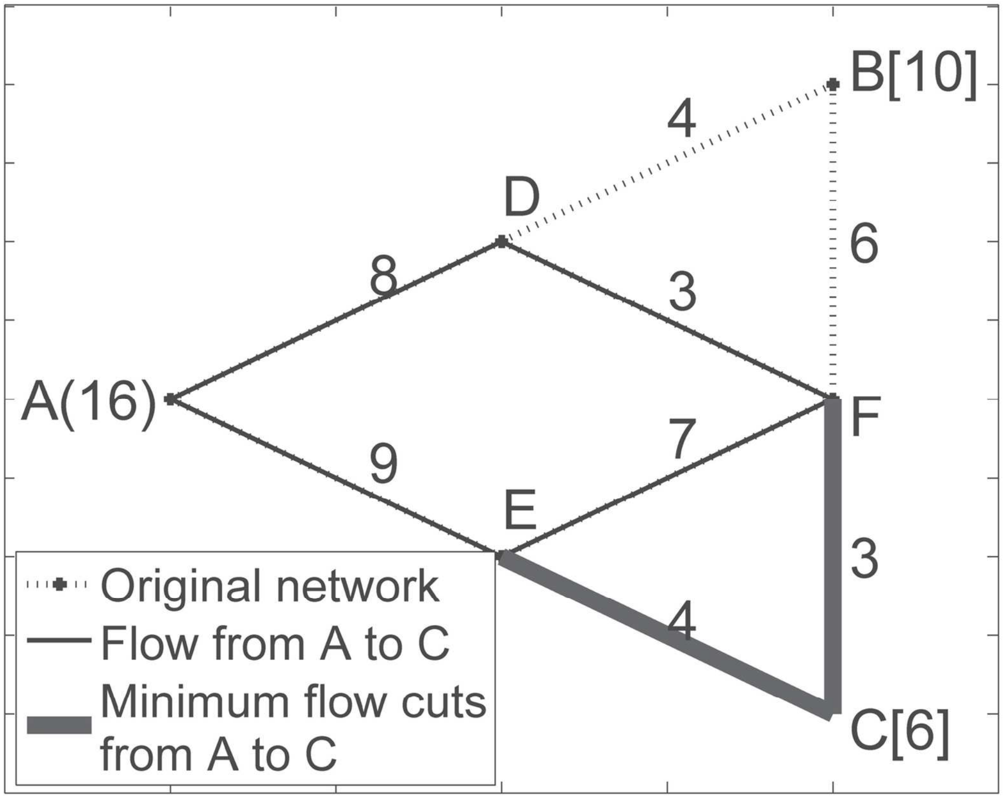
Algorithm 3 A revised cutting-plane algorithm for solving RND.

Input: An instance of problem (3).
Output: An ϵ-optimal solution to problem (3), or no feasible solution exists.
Step 0: Set $x_{ij} = 1,\ \forall (i,j) \in \mathcal{A}$. Solve $\mathrm{SP}_{\mathcal{R}}^{\mathrm{MILP}}$ in (12); obtain $\mathbf{D}$ and $\mathbf{I}$.
if $R(\mathbf{x}) = 0$ **then**
  go to **Step 1**.
else
  the instance is not feasible; **exit.**
end if
Step 1: Solve **Alt-MP1** with $\mathbf{I}, \mathbf{D}$ in (23) to obtain the current optimal solutions $\bar{\mathbf{x}}$ and $\bar{y}$.
Step 2: Fix $\mathbf{x} = \bar{\mathbf{x}}$. Then solve $\mathrm{SP}_{\mathcal{R}}^{\mathrm{MILP}}$ in (12); obtain $\mathbf{D}$ and $\mathbf{I}$.
if $R(\mathbf{x}) = 0$ **then**
  go to **Step 3**.
else
  generate feasibility cut (13) into **Alt-MP1**, and go to **Step 1**.
end if
Step 3: Solve subproblem (17) for every $k \in \mathcal{N}_-$, and choose solution $\hat{y}$ that yields the maximum objective value of all subproblems. Let $\mathbf{D}$ and $\mathbf{I}$ be optimal solutions to the subproblem corresponding to $\hat{y}$.
if $(\hat{y} - \bar{y}) \leq \epsilon \hat{y}$ **then**
  return $\bar{\mathbf{x}}$ as an ϵ-optimal and the optimal objective $\rho \sum_{(i,j)\in\mathcal{A}} c_{ij}\bar{x}_{ij} + (1-\rho)\bar{y}$
else
  generate optimality cut (22) into **Alt-MP1**, and go to **Step 1**.
end if



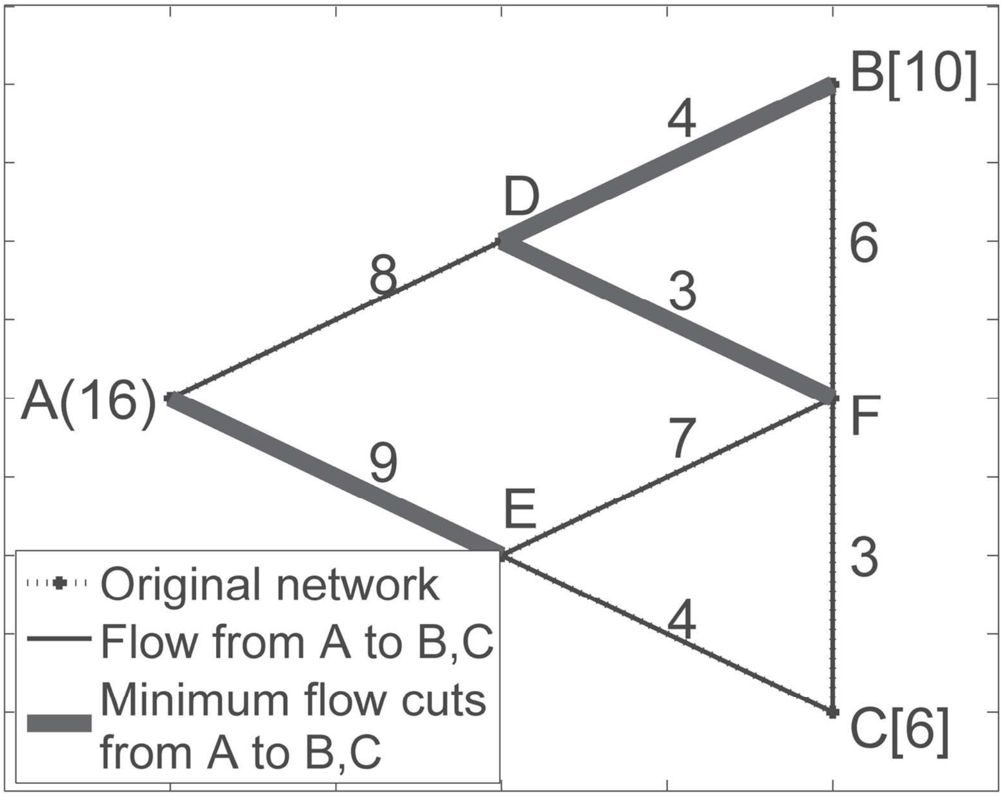
An example for demonstrating valid inequalities (33) and (34):
"Subnetwork G^B with $\mathcal{N}_-^B=\{B\}$

130x103mm (300 x 300 DPI)



An example for demonstrating valid inequalities (33) and (34):
"Subnetwork G^C with $\mathcal{N}_-^C=\{C\}$

130x103mm (300 x 300 DPI)



An example for demonstrating valid inequalities (33) and (34):
"Subnetwork G with $\mathcal{N}_-=\{B, C\}$

130x104mm (300 x 300 DPI)

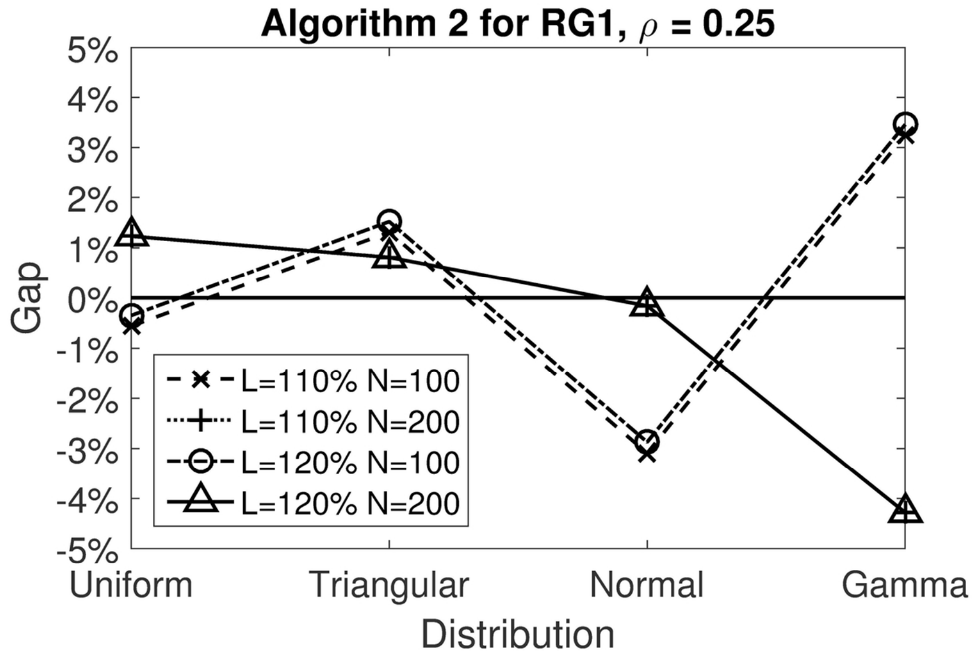


Figure 2(a) "SAA gaps in RG1 instances provided by different demand distributions, robust parameter $\mathcal L$ and sample size N for $\rho = 0.25$ and $\rho = 0.75$"

105x70mm (300 x 300 DPI)

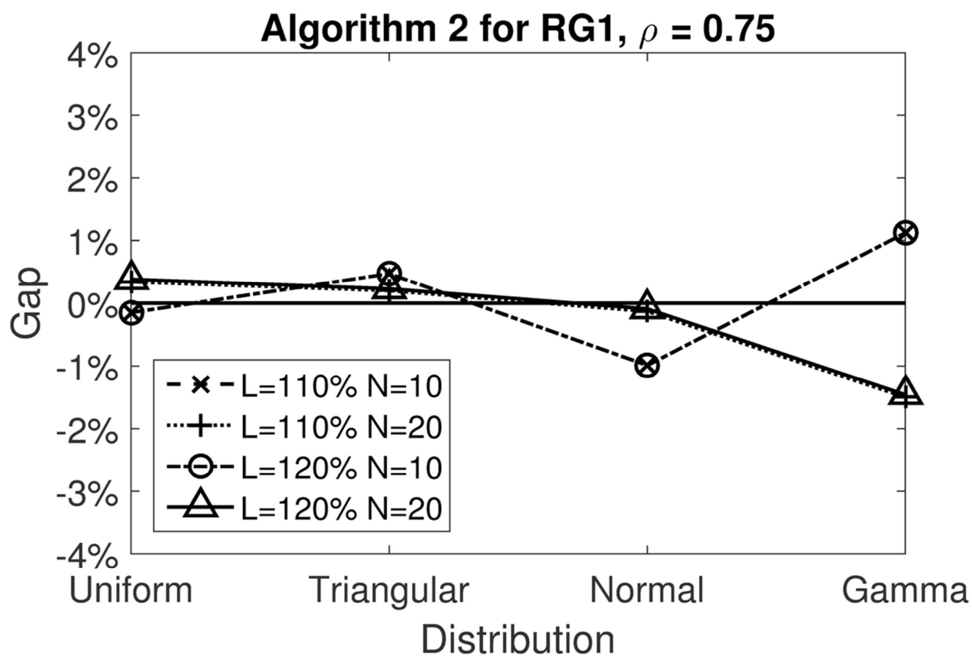


Figure 2(b) "SAA gaps in RG1 instances provided by different demand distributions, robust parameter $\mathcal L$ and sample size N for $\rho = 0.25$ and $\rho = 0.75$"

105x70mm (300 x 300 DPI)

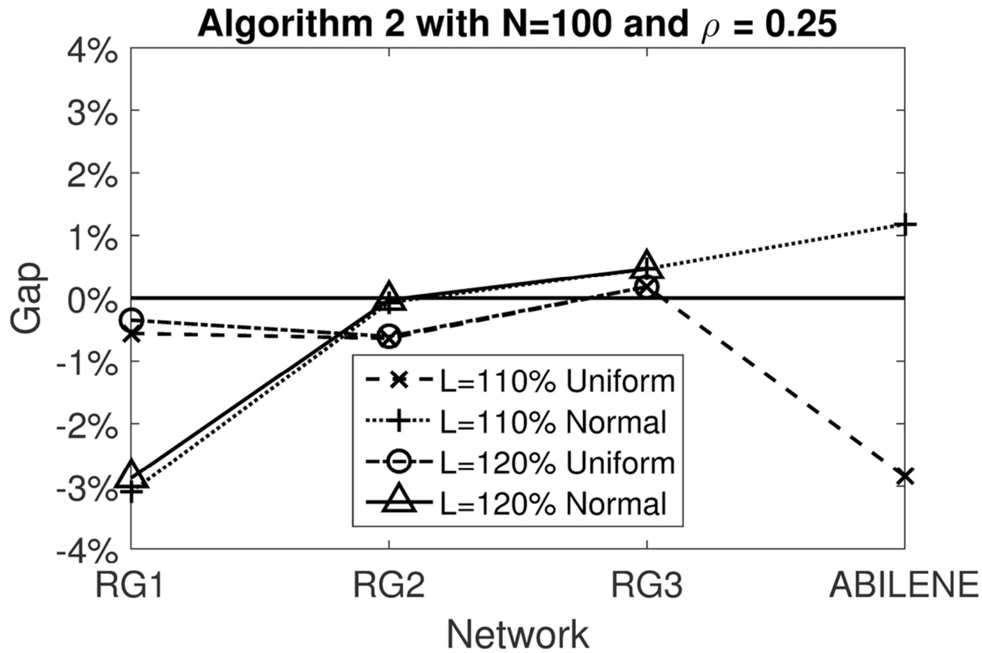


Figure 3(a) "SAA gaps in RG1, RG2, RG3 and ABILENE instances under different demand distributions and parameter $\mathcal L$ when $N =100$ for $\rho = 0.25$ and $\rho = 0.75$"

105x70mm (300 x 300 DPI)

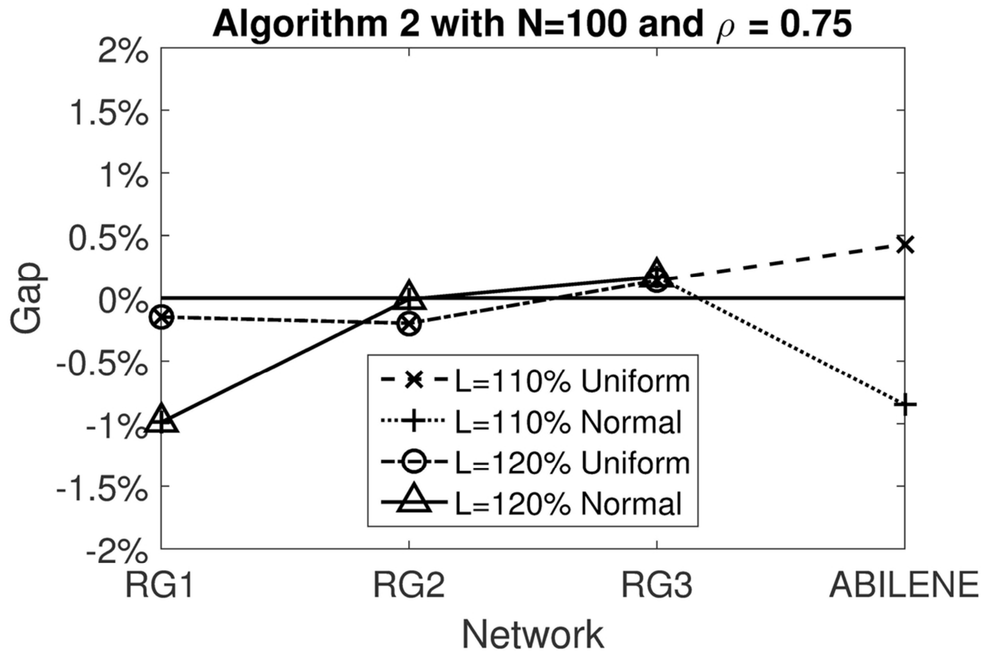


Figure 3(b): "SAA gaps in RG1, RG2, RG3 and ABILENE instances under different demand distributions and parameter $\mathcal L$ when $N =100$ for $\rho = 0.25$ and $\rho = 0.75$"

105x70mm (300 x 300 DPI)